University of California
Los Angeles

# Topics in Variational PDE Image Segmentation, Inpainting and Denoising

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

**Bing Song**

2003

The dissertation of Bing Song is approved.

Stanley Osher

Luminita Vese

Yingnian Wu

Tony F. Chan, Committee Chair

University of California, Los Angeles

2003

*To my parents and my wife*

# Table of Contents

# List of Figures

# Acknowledgments

I wish to express my sincere and deepest appreciation to my advisor, Professor Tony F. Chan, for his guidance, advice and support throughout my graduate study at UCLA. His enthusiasm, patience and intelligence have helped me achieve more in my studies. I would like to thank other members of my committee: Stanley Osher, Luminita Vese and Yingnian Wu for their help, valuable suggestions and kind encouragement. I would also like to express my gratitude to Prof. Hailiang Liu for his guidance and valuable discussions. Especially, I wish to express my sincere gratitude to Xuding Zhang for his long time support and encouragement.

I would also like to thank my colleagues for their support and discussions on various topics. Thanks goes to Haomin Zhou, Richard Tsai, Danping Peng, Sung Ha Kang, Jianhong (Jackie) Shen, Chiu-Yen Kao, Cho Hong Min, Ming-Ham (Andy) Yip, Wei Zhu, Laurent Demanet, Berta Sandberg, Mark Moelich, Jianliang Qian and Selim Esedoglu.

**Collaborations**

Chapter 2 is co authored with Tony F. Chan and submitted to *SIAM Journal on Scientific Computing*.

Chapter 3 is a joint work with Laurent Demanet.

Chapter 4 is in collaboration with Hailiang Liu.

This work was supported in part by ONR contract N00014-96-1-0277, NSF contract DMS-9973341 and NIH contract P20 MH65166.

# Vita

| | |
|---|---|
| 1973 | Born, Hubei, China. |
| 1992 | B.S. Mathematics<br>Peking University, Beijing , China |
| 1997 | M.S. Mathematics<br>Peking University, Beijing , China |
| 1999 - 2003 | Teaching Assistant and Research Assistant<br>UCLA, Los Angeles, California |

# Publications

Bing Song and Tony Chan, *A Fast Algorithm for Level Set Based Optimization* Submitted to SIAM Journal on Scientific Computing, 2003

Abstract of the Dissertation

# Topics in Variational PDE Image Segmentation, Inpainting and Denoising

by

**Bing Song**

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2003

Professor Tony F. Chan, Chair

Variational methods have been extensively used and studied in image processing in the past decade because of their flexibility in modeling and various advantages in the numerical implementation. Examples of this include image segmentation, object tracking, texture synthesis, vector field visualization, etc. This dissertation contains the study of variational methods in image segmentation, inpainting and denoising. We propose a fast algorithm for level set based optimization. The main advantage of this method is that the gradient of the functional is not needed. This enables it to be applied to broader range of optimization problems. When applying this algorithm to Chan-Vese image segmentation model, it converges extremely fast. For image inpainting, we formalize a low-level global deterministic solution for texture synthesis and image inpainting. A global energy is defined as a function of correspondence map, which is defined as linking each blank or missing pixel to the pixel where its value is taken from, in the seed image. We demonstrate why they should not be seen as procedures to sample a probability distribution on the correspondence maps. Finally, we discuss the adaptive total variation denoising. This method is a modification of the original total variation

denoising method. We prove that, after this modification, the numerical scheme for the Euler-Lagrange equation of the adaptive total variation model is stable. We show numerical experiment that this model can keep the edge of object as well as that of the TV model.

# CHAPTER 1

# Introduction

Variational and PDE methods have been widely used in image processing in the past few years. Examples include continuous mathematical morphology, shape from shading, invariant shape analysis, image segmentation, tracking, image restoration, stereo, contrast enhancement and inpainting. The basic idea is to represent an image as a $R^2$ function. This function usually satisfies a time dependent PDE that characterize the giving problem. The solution of the differential equation gives the processed image at the scale t.

There are several ways to get the PDE. One is direct derivation of the evolution equations such as the classical snakes model by Kass, Witkin and Terzopolous [KWT87], anisotropic diffusion model by Perona and Malik [Pm90], and image inpainting [BSB99]. PDE can also be obtained from variational problems. This is more common in image processing. The basic idea is to minimize an energy functional. Assume a variational approach to an image processing problem formulated as

$$arg\{min_u F(u)\},$$

where $F$ is a given energy computed over the image $u$. Let $F'(u)$ denote the Euler-Lagrange derivative. The necessary condition for $u$ to be a minimizer of $F$ is that $F'(u) = 0$, the (local) minima may be computed via the steady state solution of the equation

$$u_t = -F'(u),$$

where t is an artificial time-marching parameter. The most conventional example is the Dirichlet integral:

$$\min_u F(u) = \int_\Omega |\nabla u|^2 dxdy + \frac{\lambda}{2} \int (u - u_0)^2 dxdy,$$

which is associated with equation

$$u_t = \Delta u - \lambda(u - u_0).$$

Using variational and PDE methods in image analysis leads to modeling images in a continuous domain. This simplifies the formalism, which become grid independent and isotropic. Conversely, when the image is represented as a continuous signal, PDEs can be seen as the iteration of local filters with an infinitesimal neighborhood. This interpretation of PDE's allows one to unify and classify a number of the known iterated filters, as well as to derive new ones. Alvarez et al. [AGL93] classified all the PDE's that satisfy several axioms for image processing such as locality and causality. Another important advantage of the variational and PDE approach is the possibility of achieving high speed, accuracy, and stability with the help of the extensive results of numerical PDE methods.

Many of the PDEs used in image processing and computer vision are based on moving curves and surfaces with curvature-based velocities. In this area, the level set methods developed by Osher and Sethian [OS88] was very influential and useful. The basic idea is to represent the curves or surfaces as the zero level set of a higher dimensional hypersurface. This technique not only provides more accurate numerical implementations but also handle topological change very easily.

In next section, we discuss several variational problems arising from image segmentation, inpainting and denoising.

## 1.1 Variational and PDE Methods in Image Processing

### Image Segmentation

Image segmentation is a critical problem of early vision and has been extensively studied. Approaches to image segmentation can be roughly classified into boundary-based (or edge-based) segmentation which relies on the generation of a strength image and the prominent edges, and region-based segmentation [Par00] which relies on the homogeneity of spatially localized features and properties.

Early edge-based approaches segmentation such as Canny's edge detector use local filtering techniques. These approaches have light computational cost and stable under noise, but have difficulty in establishing the connected edge.

Kass, Witkin and Terzopolous [KWT87] proposed the classical snakes model for image segmentation, that aims at minimizing a curve-based objective functional towards a regular curve which is attracted by pixels with important gradient values. Let $\Omega$ be a bounded and open set in $R^N$, with $\partial\Omega$ its boundary. Let $u_0$ be a given image, as a bounded function defined on $\bar{\Omega}$ and with real values. For $N = 2$, $\bar{\Omega}$ is a rectangle in the plane and $u_0$ takes values between 0 and 255. Denote by $C(s) : [0, 1] \to R^2$ a piecewise $C^1[0, 1]$ planar curve. The snakes model is to minimize the following functional:

$$J_1(C) = \int_0^1 |C\prime(s)|^2 ds + \lambda \int_0^1 g(|\nabla u_0(C(s))|)^2 ds \tag{1.1}$$

The first term in (1.1) is the internal energy and control the smoothness of the curve. The second term is the external energy and attracts the contours towards the edge of the object in the image $u_0$. The function $g(|\nabla u_0|)$ called edge detector. It is strictly positive in homogenous regions, and near zeros on the edges. For

example,

$$g(|\nabla u_0|) = \frac{1}{1 + |\nabla G_\sigma * u_0|^p}, p \geq 1,$$

where $G_\sigma * u_0$, a smoother version of $u_0$, is the convolution of the image $u_0$ with the Gaussian $G_\sigma(x, y) = \sigma^{-1/2} e^{-|x^2+y^2|/4\sigma}$. Cohen [Coh91] proposed a modification of the snakes model by adding a pressure force that helps it to avoid local minima. Although these models are quite powerful, they are strongly depending on the initialization step, and need special procedures to handle topological changes on the evolving curve.

Caselles, Kimmel and Sapiro [CKS97] proposed a geodesic active contour model:

$$\inf_C J_2(c) = 2 \int_0^1 |C\prime(s)| \cdot g(|\nabla u_0(C(s))|) ds. \tag{1.2}$$

This is a problem of geodesic computation in a Riemannian space, according to a metric induced by the image $u_0$. Solving the minimization problem 1.2 consists in finding the minimal length in that matric. The level set formulation for the geodesic active contour model is:

$$\begin{cases} \frac{\partial \phi}{\partial t} = |\nabla \phi| (div(g(|\nabla u_0|) \frac{\nabla \phi}{|\nabla \phi|} + \nu g((|\nabla u_0|)), & in[0, \infty] \times R^2 \\ \phi(0, x) = \phi_0(x) & inR^2 \end{cases}$$

Here again the constant $\nu$ is added to increase the evolution speed and to attracts the curves towards the boundary and constitute an extra area-based speed. This level set formulation makes it easy to handle topological changes.

Mumford and Shah [MS89] have proposed a variational framework for image segmentation. The functional for segmentation is:

$$F^{MS}(u, C) = \nu \cdot length(C) + \lambda \int_\Omega |u - u_0|^2 dx + \int_{\Omega \setminus C} |\nabla u|^2 dx, \tag{1.3}$$

where $\lambda$, $\nu$ are positive parameters. This formulation combine three different aspects: the length of the boundary of each region, the gradient value across

the boundary pixels and the difference between the given image and a smooth version of it. The solution image $u$ obtained by minimizing this functional is formed by smooth regions $R_i$ and with sharp boundaries $C$. Several approaches have been proposed to solve this minimization problem, for instance, [CV00b] and [TYW01]. A similar approach is proposed by Chan, Vese [CV01]. This model is a special case of Mumford-Shah model in which $u$ is restrict to take piecewise constant value.

Other related works include Zhu and Yuille [ZY96], where a statistical variational approach that combines the geometrical features of a snake/ballon model and the statistical techniques of region growing. Paragio and Deriche [PD98, PD99] use a unified approach for image segmentation that incorporates probability based geodesic active region model and classical gradient based active contour techniques.

### Image Inpainting

Variational methods also have been widely used in image inpainting. Inpainting refers to the specific image restoration task of reconstructing an image with missing or damaged regions. Applications range from removing objects from a scene all the way to retouching damaged paintings and photographs. Therefore, inpainting is to fill in image information on a blank domain, based on the image information available on the rest of the image. Image denoising is different to inpainting since the regions of missing data are usually large.

Digital inpainting was first proposed by Bertalmio et.al. in [BSB99]. The basic idea is to smoothly propagate the image Laplacian in the level-lines direction. The algorithm also introduces the importance of propagating both the gradient direction (geometry) and gray-values (photometry) of the image in a

band surrounding the hole to be filled-in.

Ballester et.al. [BBC01] proposed a variational approach for image inpainting based on joint interpolation of the image grey-levels and gradient/isophotes directions, smoothly extending in an automatic fashion the isophote lines into the region of missing data.

Another approach is based on disocclusion and line continuation. In [NMS93], the authors presented a technique for removing occlusions with the goal of image segmentation. Since the region to be filled-in can be considered as occluding objects, removing occlusions is analogous to image inpainting. The basic idea suggested by the authors is to connect T-junctions at the occluding boundaries of objects with elastica minimizing curves. Masnou and Morel [MM98] extend these ideas and present a very elegant and inspiring formal variational formulation for disocclusion and a practical algorithm to implement this formulation. The algorithm fills-in by interpolating level lines from the boundary level lines.

A closely related PDE based inpainting approach is total variation (TV) minimizing inpaiting suggested by Chan and Shen [CS01]. The authors present a clear and very intuitive axiomatic approach to the problem. This approach is very similar to the classical image restoration model by Rudin, Osher and Fatemi [ROF92]. The TV inpaiting model is to minimize the following functional:

$$J_\lambda(u) = \int_{E\cup D} |\nabla u| dxdy + \frac{\lambda}{2}\int_E |u-u_0|^2 dxdy,$$

where $D$ is the inpainting domain with piecewise smooth boundary $\Gamma$, $E$ be any fixed closed domain in the complement $D^c$, so that $\Gamma$ lies in the interior of $E \cup D$ (See Fig. 1.1) and $u_0$ is the original image. Note that the constraint is only on $E$. The Euler-Lagrange equation for this energy functional is

$$-\nabla \cdot \big(\frac{\nabla u}{|\nabla u|}\big) + \lambda_e(u-u_0) = 0,$$

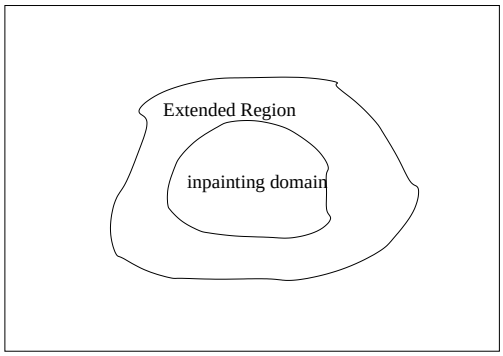



Figure 1.1: The TV inpainting model.

for all $z = (x, y) \in E \cup D$. The Lagrange multiplier $\lambda_2$ is

$$\lambda_e = \begin{cases} \lambda, z \in E \\ 0, z \in D. \end{cases}$$

If $D$ is empty, the above variational formulation is exactly the total variational denoising model. The TV inpainting is very successful if the domain $D$ is small, but sometimes it will get disconnected results even when the missing region visually looks connected or curved. To overcome this problem, Chan, Kang and Shen [CKS02] proposed the Euler's elastica and curvature based inpaintings, where the regularizing functional is

$$J_\lambda(u) = \int_{E \cup D} |\nabla u|(a + b\kappa^2)dxdy + \frac{\lambda}{2}\int_E |u - u_0|^2 dxdy,$$

Here $\kappa$ is the curvature, $\kappa = \nabla \cdot (\frac{\nabla u}{|\nabla u|})$ and $a, b$ are constants.

### Total Variation Denoising

Total Variation (TV) denoising method is a PDE-based technique that remove noise from images. The basic idea is to minimize the following functional:

$$\min_u \int_\Omega |\nabla u| dxdy + \frac{\lambda}{2}\int (u - u_0)^2 dxdy \tag{1.4}$$

Again, this is a variational problem. The TV functional does not penalize discontinuities in $u$ and thus allows us to recover the edges of the original image. The associated Euler-Lagrange equation of (1.4)is:

$$-\nabla \cdot \left( \frac{\nabla u}{|\nabla u|} \right) + \lambda(u - u_0) = 0.$$

Since $|\nabla u|$ is in the denominator, in order to avoid the singularity, it is common to use a slightly perturbed norm $|\nabla u|_\alpha = \sqrt{|\nabla u|^2 + \alpha}$, where $\alpha$ is a small positive constant, to replace $|\nabla u|$. This is equivalent to minimize the functional

$$\int_\Omega |\nabla u|_\alpha dxdy + \frac{\lambda}{2} \int (u - u_0)^2 dxdy. \tag{1.5}$$

In [AV94], it is shown that the solutions of the perturbed problems (1.5) converge to the solution of (1.4) when $\beta \to 0$.

As an important improvement of TV functional, Strong and Chan [Str97] introduced the weighted TV functional

$$TV_\alpha = \int_\Omega \alpha(x) |\nabla u(x)| dx$$

for spatially adaptive (selective) image restoration. The function $\alpha$ is chosen so that $\alpha$ is larger away from possible edges and smaller near a likely edge. Hence we allow for greater smoothing away from edges and less smoothing at the edges. Certain choices of $\alpha(x)$ were given in [CS96] and [Str97]. The numerical results are very promising. Chen and Wunderli [CW02] prove that there exists a unique solution for the weighted TV functional.

## 1.2 Contributions and Organization of This Dissertation

In this dissertation, we address several variational and PDE based image processing tasks: image segmentation, inpainting and denoising.

The usual way to solve the variational problems is to solve the corresponding Euler-Lagrange equation. But in some situation, the functional may not be differentiable. For example, if we want to segment cluttered images, $F$ can be defined as the distance between the histogram of $\phi > 0$ and that of the training set [TZ02]. To overcome this difficulty, in chapter 2, we use, instead of solving the corresponding Euler-Lagrange equation, a direct method to solve the variational problem. The main advantage is that we do not need to solve a PDE, thus have no numerical stability constraints. We simply test each point to check if the energy decreases or not when we change a point inside the level set to outside or vice versa. When we apply this method to the Chan-Vese image segmentation model [CV01], it improves the computational speed dramatically (at least 10 times). For 2-phase images, it will converge in one sweep under a suitable initial condition which usually will be satisfied. For multi-phase images, this method also converges very fast compared to PDE methods. It is easy to extend this method to higher dimensional problems such as 3-D segmentation and clustering. Another virtue of this method is that the gradient of functional $F$ is no longer required. Thus we can apply this method to a broader range of optimization problems.

Texture synthesis is important for many applications in computer graphics, vision and image processing. Unlike local deterministic inpainting, texture inpainting generally will handle larger missing region. In chapter 3, we discuss the texture inpainting problem. Our work is inspired by the algorithm of Efros and Leung [EL99], Wei and Levoy [WL00]. Their simple approach is intuitive and works surprisingly well for a wide variety of textures. We formulate a low-level variational formula for texture synthesis and image inpainting. This global energy formulation is deterministic and is defined in terms of correspondence map, which is defined as linking each blank or missing pixel to the pixel where its value is taken from, in the seed image. We demonstrate why they should not be seen

as procedures to sample a probability distribution on the correspondence maps. We therefore question the numerous claims that probability is anywhere involved at this explanatory level.

Denoising is another kind of image restoration. The Total Variation (TV) denoising model has been extensively studied since it is proposed by Rudin, Osher and Fetami [ROF92]. This model can effective remove the noise in image while still keep the sharp edges. In chapter 4, we propose an alternative model in order to still keep the sharp edges for the real edge and avoid the staircase effect in the smooth region. The novelty of the new method is to use an adaptive $L^p$ norm, $1 < p < 2$, which will lead to a well posed Euler-Lagrange equation. We prove that after this modification, the gradient descent equation of the functional is convergent and stable. We show numerical experiment that this model can keep the edge of object as well as that of the TV model. It also reduces the staircase effect in the TV model.

# CHAPTER 2

# Fast Algorithm for Level Set Based Optimization

## 2.1 Introduction

Variational methods have been extensively used and studied in image processing in the past decade because of their flexibility in modeling and various advantages in the numerical implementation. Examples of this include image segmentation [CV00a, CV01, KLM94, PD98], object tracking [PD00], texture synthesis and vector field visualization [BCO01]. The basic idea of variational methods is to minimize a cost or energy functional. This functional generally will depend on the features of the image. The classical way to solve the minimization problem is to solve the corresponding Euler-Lagrange equation. This PDE based method sometimes is not very efficient because of numerical stability constraints.

In this chapter, we present a new algorithm for solving a kind of optimization problems that can be formulated by level sets. That is, our algorithm is defined for:

$$\min_{\phi} F(H(\phi)), \tag{2.1}$$

where $H$ is the Heaviside function, $F$ is any functional dependent on $H(\phi)$ and $\phi$ is the level set function. In particular, the Chan-Vese model [CV00a] is in this form. This minimization problem is usually solved by solving the Euler-Lagrange

equation:

$$\phi_t = -\nabla_\phi F(H(\phi)). \tag{2.2}$$

The prerequisite for this method is that $F$ must be differentiable with respect to $\phi$. But for many problems, this may not be true. For example, if we want to segment cluttered images, $F$ can be defined as the distance between the histogram of $\phi > 0$ and that of the training set [TZ02]. To overcome this difficulty, in this paper, we use, instead of solving the corresponding Euler-Lagrange equation, a direct method to solve the variational problem. The main advantage is that we do not need to solve a PDE, thus have no numerical stability constraints. We simply test each point to check if the energy decreases or not when we change a point inside the level set to outside or vice versa. When we apply this method to the Chan-Vese image segmentation model [CV01], it improves the computational speed dramatically (at least 10 times). For 2-phase images, it will converge in one sweep under a suitable initial condition which usually will be satisfied. For multi-phase images, this method also converges very fast compared to PDE methods. It is easy to extend this method to higher dimensional problems such as 3-D segmentation and clustering. Another virtue of this method is that the gradient of functional $F$ is no longer required. Thus we can apply this method to a broader range of optimization problems.

The possibility of obtaining a fast algorithm for the Chan-Vese model by solving ordinary differential equation was pointed out to the second author by Fedkiw in [Fed01]. This has led to the recent work of Gibou and Fedkiw in [GF02], which is similar, but not identical, to our current work.[1] The methods are similar in the sense that they both are designed to speed up the computation of the Chan-Vese model, and they both exploit the fact that we only need the

[1]Apparently, we were working independently and were not aware of each other's investigation.

sign of $\phi$ but not its value. They are different in their specific approaches. The method in [GF02] is motivated by considering large time steps for a simplified Euler-Lagrange equation of the Chan-Vese model by ignoring the length regularization term. The regularization is then put back in by a subsequent anisotropic diffusion process. Our approach is a general level set based optimization framework, with the Chan-Vese model as a particular application. We use the values of the objective function directly, without the need for its gradient or the corresponding Euler-Lagrange equation, to determine the sign of $\phi$. Our model also allows the full Chan-Vese model with the length term included. We will have a more detailed comparison later in the paper.

The remainder of this chapter is organized as follows. In section 2.2, we briefly review the level set methods and give our new algorithm to solve the optimization problem (2.1). In section 2.3, we review the Chan-Vese model and analyze its properties in section 2.4. These properties are for the Chan-Vese model only. In section 2.5 we apply the algorithm to various extension of Chan-Vese model. Numerical implementation and experimental results are given in section 2.6, and we end the paper with a brief concluding section.

## 2.2 The Level Set Based Optimization

We begin by reviewing the standard level set method, then we give a new method to solve the optimization problem (2.1) and set the conventions that will be used throughout the paper.

The level set method, as initiated in [OS88], has been widely used in many areas including computational physics, image processing and computer graphics. The idea of the level set formulation is that it represents the front as the zero

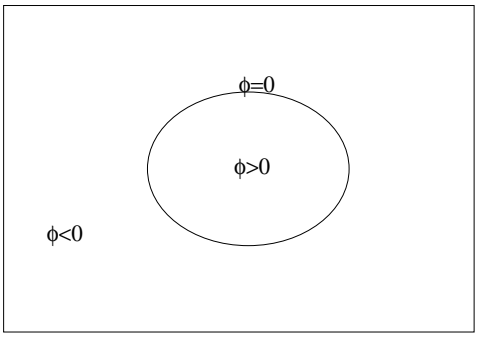



Figure 2.1: Zero contour of $\phi$ representing the front $\Gamma$

level set of a function defined in a higher dimensional space.

Consider a closed moving interface $\Gamma(t)$ in $R^n$. Let $\Omega(t)$ be the region (possibly multi-connected) that $\Gamma(t)$ encloses. We associate with $\Omega(t)$ an auxiliary function $\phi(x,t)$, called the level set function, which is Lipschitz continuous and satisfies:

$$\begin{cases} \phi(x,t) > 0 & \text{for } x \in \Omega \\ \phi(x,t) = 0 & \text{for } x \in \partial\Gamma \\ \phi(x,t) < 0 & \text{for } x \in \Omega^c, \end{cases} \tag{2.3}$$

where $x \in R^n, t \in R^+$. See Fig. 2.1

Conversely, if we know $\phi$, we can locate the interface by finding the zero level set of $\phi$. That is $\Gamma(t) = \{x : \phi(x,t) = 0\}$. So moving the interface is equivalent to update $\phi$, which can be done by solving a Hamilton-Jacobi equation. The equation can be derived as follows.

suppose $x(t)$ is a particle trajectory on the interface $\Gamma(t)$ moving with velocity $\vec{u} = \dot{x}(t)$. By definition, we have $\phi(x(t),t) = 0$. Differentiating with respect to $t$, we get

$$\phi_t + \vec{u} \cdot \nabla\phi = 0. \tag{2.4}$$

By projecting the velocity $\vec{u}$ onto the normal direction $\vec{n}$ of the interface, Equation

(2.4) becomes

$$\phi_t + u_n|\nabla\phi| = 0. \tag{2.5}$$

This is the level set equation given by Osher and Sethian [OS88]. Typically, the velocity $\vec{u}$ or $u_n$ is giving on the interface. It might be a function of space variable $x$, time $t$, the normal direction $\vec{n}$, the local mean curvature $\kappa$, or some global quantities like the Hausdorff measure $|\Gamma(t)|$ of the interface or the Lebesgue measure $|\Omega(t)|$ of $\Omega(t)$, or some external physics to which the motion of $\Gamma(t)$ is coupled. In terms of $\phi$, we have the following simple facts:

$$\hat{n} = -\frac{\nabla\phi}{|\nabla\phi|}, \tag{2.6}$$

$$|\Gamma(t)| = \int |\nabla H(\phi)|dx, \tag{2.7}$$

$$|\Omega(t)| = \int H(\phi)dx, \tag{2.8}$$

where $\hat{n}$ is the outward normal of the interface $\Gamma(t)$, $H(\phi)$ is the 1D Heaviside function which takes 0 for $\phi < 0$ and 1 otherwise. In 2D, $|\Gamma(t)|$ is simply the arc length of $\Gamma(t)$ and $|\Omega(t)|$ the area of $\Omega(t)$, while in 3D, $|\Gamma(t)|$ is the surface area of $\Gamma(t)$ and $|\Omega(t)|$ the volumn of $\Omega(t)$.

If we know $\phi$, we can locate the interface by finding the zero level set of $\phi$. That is, $\Gamma(t) = \{x : \phi(x,t) = 0\}$. So moving the interface is equivalent to updating $\phi$, which can be done by solving a Hamilton-Jacobi type equation such as (2.2). This is usually slow because of the CFL condition. When solving (2.1), if we really need the value of level set $\phi$, we have to solve (2.2). But for some special cases, for example, the Chan-Vese segmentation model, we do not need the value of $\phi$, but only its sign. From the optimization point of view, this opens to us the possibility to use other direct methods to solve the minimization problem like (2.1). Based on this observation, we give a new direct algorithm to solve (2.1).

The outline of the algorithm is as follows:

**Step 1.** Initialize. Construct an initial partition, one part for $\phi > 0$, one part for $\phi < 0$ and compute the value of $F$ according to $\phi$.

**Step 2.** Advance. For each point $x$ in image, if the energy $F$ decreases when we change $\phi(x)$ to $-\phi(x)$ , then update this point by $\phi(x) = -\phi(x)$, otherwise, $\phi(x)$ remains unchanged. We sweep the pixels in some prescribed order. For example, in image segmentation, we can sweep the pixels row by row. We can use either Gauss-Seidel or Jacobi iteration in each sweep.

**Step 3.** Repeat the step 2 until the energy $F$ remains unchanged.

In step 1, there are several ways to set the initial value of $\phi$. For example, we can use signed distance to the zeros level set as initial value or just let $\phi = 1$ inside the level set and $\phi = -1$ outside. Which one will be used depends on the expression of $F$. Note that after initialization, we only change the sign of $\phi$ in each sweep. There is no need to compute derivatives and $F$ need not be differentiable. On the other hand, in the above heuristic approach, we can not guarantee $F$ has a unique solution. We also do not know whether the algorithm can converge even to a local minimum.

To illustrate this algorithm, we apply it to the Chan-Vese image segmentation model [CV01] and see how it improves the convergence speed. We also prove some properties of this method.

## 2.3 Application to the Chan-Vese Segmentation Model

The Chan-Vese image segmentation model is a variational model for 2-phase image segmentation. The basic idea is to look for a particular partition of a

given image into two regions, one representing the objects to be detected and one representing the background. Assuming that the image $u_0$ is a 2-phase image with piecewise constant values $u_0^i$ and $u_0^o$ and that the object to be detected is represented by the value $u_0^i$. Let $C_0$ denote the boundary of the object. The "fitting energy" is defined as :

$$F_1(C) + F_2(C) = \int_{inside(C)} |u_0 - c_1|^2 + \int_{outside(C)} |u_0 - c_2|^2,$$

where $C$ is any other variable curve, and the constants $c_1, c_2$ are the averages of $u_0$ inside and outside of $C$ respectively. The fitting energy will be minimized if $C = C_0$. In the Chan-Vese model, they also have a regularizing term, such as the length of $C$ and the area inside $C$ to control the smoothness of the boundary. Therefore, the energy $F(C, c_1, c_2)$ is define by:

$$F(C, c_1, c_2) = \mu \cdot (length(C)) + \lambda_1 \int_{inside(C)} |u_0 - c_1|^2 + \lambda_2 \int_{outside(C)} |u_0 - c_2|^2 \tag{2.9}$$

If we use the level set to represent $C$, that is, $C$ is the zero level set of a Lipschitz function $\phi : R^2 \to R$, then we can replace the unknown variable $C$ by the unknown variable $\phi$, and the energy functional $F(C, c_1, c_2)$ can be written as:

$$\begin{aligned} F(H(\phi), c_1, c_2) = \mu\big(\int_\Omega |\nabla H(\phi)|\big) + \lambda_1 \int_\Omega |u_0 - c_1|^2 H(\phi) dx \\ + \lambda_2 \int_\Omega |u_0 - c_2|^2 (1 - H(\phi)) dx. \end{aligned} \tag{2.10}$$

where $c_1$, $c_2$ are also functions of $H(\phi)$. Note that $F$ is in the form of (2.1). In equation (2.10), the two fitting terms are easy to compute directly. We can approximate $\int |\nabla H(\phi)| dx$ by:

$$\sum_{i,j} \sqrt{(H(\phi_{i+1,j}) - H(\phi_{i,j}))^2 + (H(\phi_{i,j+1}) - H(\phi_{i,j}))^2},$$

where $\phi_{i,j}$ is the value of $\phi$ at the $i, j$th pixel. The summand can only take the values 0, 1 or $\sqrt{2}$, depending on whether the 3 distinct pair of points from the

set $\{ \phi_{i,j}, \phi_{i+1,j}, \phi_{i,j+1}\}$ belong to the same or different regions. Thus the length term can be easily computed knowing only $H(\phi)$, and there is no need to know $\phi$. This computed value can be interpreted as the discretized length of zero level set. Note that to apply our algorithm, we do not need $F$ differentiable in (2.10), which would have necessitied $\delta(\phi)$ in Euler-Lagrange equation of (2.10).

To solve the minimization problem (2.10), the usual approach is to derive its Euler-Lagrange equation:

$$\frac{\partial \phi}{\partial t} = \delta(\phi)(div(\mu \frac{\nabla \phi}{|\nabla \phi|} - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2), \tag{2.11}$$

where

$$c_1(\phi) = \frac{\int_\Omega u_0 H(\phi) dxdy}{\int_\Omega H(\phi) dxdy},$$
$$c_2(\phi) = \frac{\int_\Omega u_0 (1 - H(\phi)) dxdy}{\int_\Omega (1 - H(\phi)) dxdy}.$$

The CFL condition for this equation is:

$$\frac{\Delta t}{\Delta x^2} \leq c \frac{\nabla \phi}{\delta(\phi)}$$

Because the time step should be very small, it needs a lot of iterations to converge. We can also use implicit iterative methods, such as those used in the original paper [CV00a]. In this case, the time step is not as restricted as in explicit time marching.

Our algorithm for the Chan-Vese model is:

**1.** Give any initial partition of the image, set $\phi = 1$ for one part and $\phi = -1$ for another part.

**2.** Assume that the value of current pixel $(i, j)$ is $u$, $c_1$ and $c_2$ are averages for $\phi = 1$ and $\phi = -1$ respectively, $m$ and $n$ are number of pixels for $\phi = 1$

and $\phi = -1$. If $\phi(i,j) = 1$, then compute the difference between the new and the old energy:

$$\Delta F_{12} = (u - c_2)^2 \frac{n}{n+1} - (u - c_1)^2 \frac{m}{m-1}.$$

If $\Delta F_{12} < 0$, then change $\phi(i,j)$ from 1 to -1.

And similarly for the $\phi(i,j) = -1$ case. If we consider the length term, then the change of the length is easy to compute since only four neighbor points will be affected when we change the value of a point.

**3.** Repeat the step 2 until the total energy $F$ remains unchanged.

## 2.4 Analysis of the Algorithm for Chan-Vese Model

When we apply our new algorithm to the Chan-Vese model, it always converges in a finite number of sweeps (usually less than 10). If we do not consider the length term ($\mu = 0$), it even converges in less than 5 sweeps. This leads us to analyze a simplified form of the algorithm that leaves out the regularization in the Chan-Vese model.

Considering a two phase image, the object is represented by $A$ (maybe multi-connected), the background is $B$, the corresponding value for $A$ and $B$ is $a$ and $b$, Given an initial partition $\phi > 0$ and $\phi < 0$, denoted by $\phi_1$ and $\phi_2$. Assume there are $m$ points in $\phi_1$ and $n$ points in $\phi_2$. Let $c_i$, $F_i$ be the average and energy for $\phi_i$, $i = 1, 2$. Assume a point $P \in \phi_1$ with value $u$. If we change $P$ to from $\phi_1$ to $\phi_2$, let $\tilde{c_1}$, $\tilde{c_2}$ be the new average for $\phi_1$ and $\phi_2$, respectively, and $\tilde{F_1}, \tilde{F_2}$ be the

new energy for $\phi_1$ and $\phi_2$. Then we can easily calculate:

$$\tilde{c_1} = c_1 + \frac{c_1 - u}{m - 1}$$
$$\tilde{c_2} = c_2 - \frac{c_2 - u}{n + 1}$$
$$\tilde{F_1} = F_1 - (u - c_1)^2 \frac{m}{m - 1}$$
$$\tilde{F_2} = F_2 + (u - c_2)^2 \frac{n}{n + 1}.$$

The difference between the new energy and old energy is:

$$\Delta F_{12} = (u - c_2)^2 \frac{n}{n + 1} - (u - c_1)^2 \frac{m}{m - 1}.$$

Similarly, if $P$ change from $\phi_2$ to $\phi_1$, the change of energy is:

$$\Delta F_{21} = (u - c_1)^2 \frac{m}{m + 1} - (u - c_2)^2 \frac{n}{n - 1}.$$

Now, we have:

**Lemma 1.** *If $A \subseteq \phi_1$ and $\frac{n}{n+1} > \frac{(m-A)^2}{m^2} \frac{m}{m-1}$, where $A$ represent the totality of points in object $A$, then, after one sweep, the algorithm will converge to global minimum. The object is either $\phi > 0$ or $\phi < 0$.*

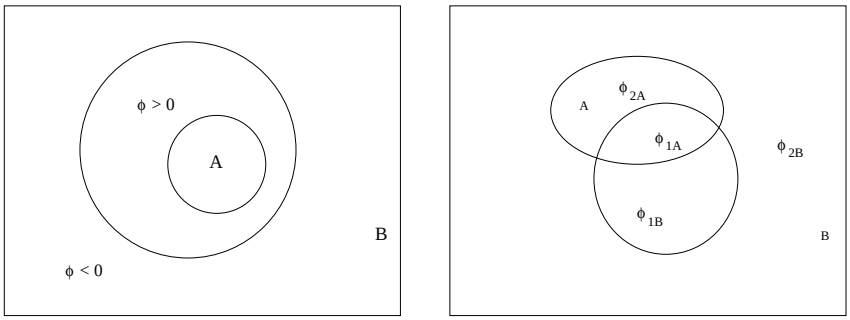
*Proof.* Assume $a < b$, then $c_2 = b, a < c_1 < b$. We only need to consider every possible case. See Fig. 2.2(a). Assume $P \in A$. If we change $P$ to $\phi_2$, the change of energy will be:

$$\begin{aligned}\Delta F_{12} &= \frac{n}{n + 1}(a - b)^2 - (a - c_1)^2 \frac{m}{m - 1} \\ &= (\frac{n}{n + 1} - (\frac{m - A}{m})^2 \frac{m}{m - 1})(a - b)^2.\end{aligned}$$

By assumption, this is positive, so $P$ will not change.

If $P \in B \cap \phi_1$, then

$$\Delta F_{12} = -(b - c_1)^2 \frac{m}{m - 1} < 0,$$




(a) (b)

Figure 2.2: different initial condition, (a), the object is completely contained in $\phi > 0$, (b), the object A has nonempty intersection with $\phi > 0$ and $\phi < 0$

so $P$ will change to $\phi_2$.

If $P \in B \cap \phi_2$, then if change $P$ from $\phi_2$ to $\phi_1$,

$$\Delta F_{21} = (b - c_1)^2 \frac{m}{m+1} > 0,$$

therefore, $P$ will not change.

So, after one sweep, all points are computed correctly. The object is represented by $\phi > 0$. This complete our proof. □

*Remark 1.* For $A \subseteq \phi_2$, the proof is the same. In practice, $\frac{n}{n+1} \approx 1, \frac{m}{m-1} \approx 1$, so if $\phi_1$ is close enough to $A$, then $m - A << m$, and the condition in lemma 1 will be satisfied.

*Remark 2.* In the above proof, we use Jacobi iteration. The sweeping order is not important since we do not use the improved values of $F$ until after a complete sweep. We can use any sweeping order in step 2. If we use the Gauss-Seidel iteration, that is, for each point, we use the most recent value, we still have the same result. To prove this, we only need to show that the condition in lemma 1 will be satisfied for each point. Given a point $P$, if $P \in A$ or $P \in B \cap \phi_2$, then

from the proof of lemma 1, $\phi(P)$ will not change. If $P \in B \cap \phi_1$, then $P$ will change from $\phi_1$ to $\phi_2$. Note that after this calculation, $m = m - 1$, $n = n + 1$, so the condition in lemma 1 is still satisfied. This completes our proof.

*Remark 3.* Strictly speaking, our method does not use level set evolution, We just use level set to formulate our problem.

We now consider the other case, i.e. $A$ and $B$ can not be completely contained in $\phi > 0$ or $\phi < 0$, we have:

**Lemma 2.** *Let $\phi_{1A} = A \cap \phi_1$, $\phi_{2A} = A \cap \phi_2$, $\phi_{1B} = B \cap \phi_1$, $\phi_{2B} = B \cap \phi_2$, we assume none of them are empty, i.e., $A$ and $B$ can not be completely contained in $\phi_1$ or $\phi_2$. See Fig.2.2(b). If $\phi_{1A}$ does not change from $\phi_1$ to $\phi_2$, then $\phi_{2A}$ must change sign. Similar for $\phi_{1B}$ and $\phi_{2B}$.*

*Proof.* Since $\phi_{1A}$ does not change sign, then

$$\Delta F_{12} = \frac{n}{n+1}(a - c_2)^2 - (a - c_1)^2 \frac{m}{m-1} \geq 0,$$

so we have

$$\frac{(a - c_2)^2}{(a - c_1)^2} \geq \frac{m}{m-1}\frac{n+1}{n} > 1.$$

Now, if $\phi_{2A}$ does not change from $\phi_2$ to $\phi_1$, then

$$\Delta F_{21} = \frac{m}{m+1}(a - c_1)^2 - (a - c_2)^2 \frac{m}{m-1} \geq 0,$$

so we have

$$\frac{(a - c_2)^2}{(a - c_1)^2} \leq \frac{m}{m+1}\frac{n-1}{n} < 1.$$

This is a contradiction. □

This means if one part of $A$ does not change, the other part of $A$ must change to the same sign.

*Remark 4.* It is possible that $\phi_{1A}$ and $\phi_{1B}$ change sign at the same time. In this case, it must satisfy the following condition:

$$\frac{n-1}{n}\frac{m}{m+1} \leq \frac{(a-c_2)^2}{(a-c_1)^2} \leq \frac{m}{m-1}\frac{n+1}{n}. \tag{2.12}$$

For example, if $c_1 = c_2$, then this condition will be satisfied. In the next section, we give an example satisfying this condition, and the algorithm ceases to work. This condition means that $c_1$ and $c_2$ are very close. In general, if the object is not similar to the background and the initial partition is close to object, then condition (2.12) will not be satisfied and the algorithm will converge very quickly.

Now we have our main theorem:

**Theorem 1.** *If the condition in inequality (2.12) is not satisfied, and $|c_1 - c_2| > c$, where $c$ is some constant, then the algorithm ( with $\mu = 0$ ) with either Jacobi or Gauss-Seidel converges in one sweep for a 2-phase image.*

*Proof.* First we consider Jacobi type iteration. Since inequality (2.12) is not satisfied, without loss of generality, we assume $\frac{(a-c_2)^2}{(a-c_1)^2} \geq \frac{m}{m-1}\frac{n+1}{n}$. We further assume $c_2 > c_1$, $a < b$. Then $a < c_1 < c_2 < b$. If $A$ or $B$ is completely contained in $\phi_1$ or $\phi_2$, then by lemma 1, it will converge in one sweep. Otherwise, both $A, B$ have intersection with $\phi_1$ and $\phi_2$. Since $\frac{(a-c_2)^2}{(a-c_1)^2} \geq \frac{m}{m-1}\frac{n+1}{n}$, so

$$\Delta F_{12} = \frac{n}{n+1}(a-c_2)^2 - (a-c_1)^2\frac{m}{m-1} \geq 0,$$

which means $\phi_{1A}$ does not change sign. By lemma 2, $\phi_{2A}$ will change sign from

$\phi_2$ to $\phi_1$. So $A \in \phi_1$. Now consider a point $P$ in $\phi_{2B}$. If it changes to $\phi_1$, then

$$\Delta F_{21} = \frac{n}{n+1}(b-c_1)^2 - (b-c_2)^2\frac{m}{m-1} \geq 0.$$

This means $\phi_{2B}$ will not change sign. By lemma 2, $\phi_{1B}$ will change sign from $\phi_1$ to $\phi_2$ and we have $B \in \phi_2$. If we use Jacobi iteration, then, after one sweep, $A \in \phi_1$ and $B \in \phi_2$. If we use Gauss-Seidel iteration, as the argument in remark 2 shows, each time when we change a point, the condition in the theorem is still satisfied. This completes our proof. □

*Remark 5.* In practice, we do not expect one sweep convergence by just considering local change since the Chan-Vese model is a global model. In this model, the only global sharing information is the average. In our algorithm, when we change a point, we know how it affects the average. That's why it is possible to converge in one sweep.

Generally, it is easy to have initial conditions that satisfies the hypothesis for theorem 1. Our experiments also show that it is correct for 2-phase images. For multi-phase images, we no longer have finite convergence result of theorem 1, but we still can expect fast convergence.

In practice, when we apply our algorithm to the Chan-Vese model, we have several choices. The first one is to directly apply the algorithm to the model with the length term included. Or we can first consider $\mu = 0$, then followed by $\mu > 0$ to have the full effect of regularization. Another choice is to consider $\mu = 0$, followed by a PDE-based algorithm. We will give an example in the section 2.6.

## 2.5 Application to Other Optimization Problems

### 2.5.1 Piecewise Linear Chan-Vese model

A simple extension to the Chan-Vese model is to use linear approximation instead of constant [TZ02, Ves02]. This model is more appropriate when the image has region of linear shading instead of piecewise constant intensities. We consider a 2-phase image segmentation, each phase can be approximated by a linear function. Thus, using the level set representation, the functional can be represented as:

$$\begin{aligned} F(H(\phi)) = \mu\big(\int_\Omega |\nabla H(\phi)|\big) + \lambda_1 \int_\Omega |u_0 - a_0 - a_1 x - a_2 y|^2 H(\phi) dxdy \\ + \lambda_2 \int_\Omega |u_0 - b_0 - b_1 x - b_2 y|^2 (1 - H(\phi)) dxdy, \end{aligned} \tag{2.13}$$

where $a_i$, $b_i$ , $i = 0, 1, 2$, are coefficients of a linear function which depend on $H(\phi)$. The $a_i$ can be computed by linear system:

$$\frac{\partial}{\partial a_i} \int_\Omega |u_0 - a_0 - a_1 x - a_2 y|^2 H(\phi) dxdy = 0, i = 0, 1, 2.$$

Similarly for $b_i$. It is as easy to evaluate $F$ and apply our algorithm as in the Chan-Vese model. But in this model, updating $a_i$ and $b_i$ is not so easy when we change $\phi$ of a point. So it is better to use Jacobi iteration than Gauss-Seidel iteration. The experiment result will be given in next section.

### 2.5.2 Multiphase Chan-Vese Model

The Chan-Vese model is a special case of Mumford-Shah energy. It is natural to extend the two-phase Chan-Vese model to the general case of piecewise-constant optimal approximations. In the classical multiphase approaches by level sets, as in (Zhao, Chan, Merriman and Osher, 1996), each phase is represented by a level set function. A region or phase $\Omega_i$ is therefore defined by:

$$\Omega_i = \{(x, y) \in \Omega : \phi_i(x, y) \geq 0\},$$

and the boundaries between phases are defined by the union of the zeros level sets of $\phi_i$, each boundary being represented twice. In order to prevent vacuum and overlap (i.e, to ensure $\Omega_i$ disjoint and $\cup_i \Omega_i = \Omega$), we must add the following constraint:

$$\sum_{i}^{n} H(\phi_i) = 1 forall(x, y) \in \Omega.$$

This condition has to be reinforced numerically at each step. If many phases are involved, then this representation is computationally expensive.

Another level set representation for multiphase images is introduced by Chan, Vese in [CV00a]. The idea is using $\log n$ level set functions to represent $n$ phases or segments. This formulation automatically removes the problem of vacuum and overlap, because the partition is a disjoint decomposition and covering of the domain $\Omega$ by definition. For example, we can use two level set functions $\phi_1$ and $\phi_2$ to represent a 4 phase image which consists of four regions: $\{\phi_1 > 0, \phi_2 > 0\}$, $\{\phi_1 > 0, \phi_2 < 0\}$, $\{\phi_1 < 0, \phi_2 > 0\}$, $\{\phi_1 < 0, \phi_2 < 0\}$ (see Fig. 2.3).

Therefore, the energy to be minimized is given by:

$$F_n(C, \Phi) = \sum_{1 \leq I \leq = 2^m} \int_{\Omega} |u_0 - c_I| \chi_I dxdy + \sum_{1 \leq i \leq m} \nu \int_{\Omega} |\nabla H(\phi_i)|. \tag{2.14}$$

Here, the set of curves $C$ is represented by the union of the zero level sets of the functions $\phi_i$. For the purpose of illustration, let us write the above energy for

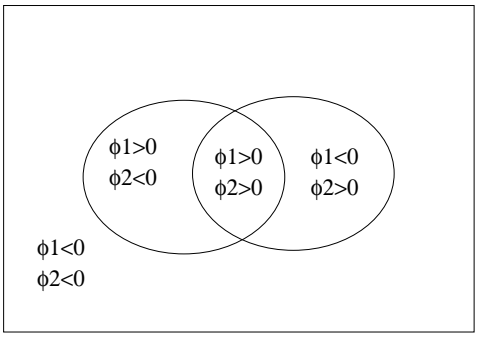



Figure 2.3: Two curves given by $\phi_1 = 0$ and $\phi_2 = 0$, partition the domain into four regions:$\{\phi_1 > 0, \phi_2 > 0\}$, $\{\phi_1 > 0, \phi_2 < 0\}$,$\{\phi_1 < 0, \phi_2 > 0\}$, $\{\phi_1 < 0, \phi_2 < 0\}$

$n = 4$ phase:

$$\begin{aligned} F_4(c,\phi) = &\int_\Omega |u_0(x,y) - c_{11}|^2 H(\phi_1)H(\phi_2)dxdy \\ &+ \int_\Omega |u_0(x,y) - c_{10}|^2 H(\phi_1)(1 - H(\phi_2))dxdy \\ &+ \int_\Omega |u_0(x,y) - c_{01}|^2 (1 - H(\phi_1))H(\phi_2)dxdy \\ &+ \int_\Omega |u_0(x,y) - c_{00}|^2 (1 - H(\phi_1))(1 - H(\phi_2))dxdy \\ &+ \nu \int_\Omega |\nabla H(\phi_1)| + \nu \int_\Omega |\nabla H(\phi_2)| \end{aligned} \tag{2.15}$$

, where $c = (c_{11}, c_{10}, c_{01}, c_{00})$ is a constant vector, and $\Phi = (\phi_1, phi_2)$.

With these notations, the image function $u$ can be expressed as:

$$\begin{aligned} u = &c_{11}H(\phi_1)H(\phi_2) + c_{10}H(\phi_1)(1 - H(\phi_2)) \\ &+ c_{01}(1 - H(\phi_1))H(\phi_2) + c_{00}(1 - H(\phi_1))(1 - H(\phi_2)). \end{aligned}$$

The Euler-Lagrange equations obtained by minimizing (2.15) with respect to

$c$ and $\Phi$ are:

$$\begin{cases} c_{11} = mean(u_0) in \{\phi_1 > 0, \phi_2 > 0\} \\ c_{10} = mean(u_0) in \{\phi_1 > 0, \phi_2 < 0\} \\ c_{01} = mean(u_0) in \{\phi_1 < 0, \phi_2 > 0\} \\ c_{00} = mean(u_0) in \{\phi_1 < 0, \phi_2 < 0\}, \end{cases}$$

$$\begin{aligned} \frac{\partial \phi_1}{\partial t} = \delta_\varepsilon(\phi_1)\Big\{ div(\frac{\nabla \phi_1}{|\nabla \phi_1|}) - \Big[\Big((u_0 - c_{11})^2 - (u_0 - c_{01})^2\Big) H(\phi_2) \\ - \Big((u_0 - c_{10})^2 - (u_0 - c_{00})^2\Big)(1 - H(\phi_2))\Big]\Big\}, \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \phi_2}{\partial t} = \delta_\varepsilon(\phi_2)\Big\{ div(\frac{\nabla \phi_2}{|\nabla \phi_2|}) - \Big[\Big((u_0 - c_{11})^2 + (u_0 - c_{01})^2\Big) H(\phi_1) \\ - \Big((u_0 - c_{10})^2 + (u_0 - c_{00})^2\Big)(1 - H(\phi_1))\Big]\Big\} \end{aligned}$$

Again, we need to solve two PDE's for $\phi_1$ and $\phi_2$. It needs a lot of iteration as in the 2-phase Chan-Vese model. Now, we apply our fast algorithm to this minimization, we only need to test how the energy will change when we change a point from one region to other three regions, then choose the region that has minimum energy. The numerical results are shown in next section.

### 2.5.3 Texture Segmentation

Texture segmentation is one of the most important techniques for image analysis, understanding and interpretation. The task of texture segmentation is to partition the image into a number of regions that each region has the same texture properties. This can also be viewed as the problem of accurately extracting the borders between different texture regions. There are several problems specific to texture segmentation: when two textures regions have same intensities, it is

very difficult for standard segmentation models to segment them. To overcome this difficulty, one of the common technique is to use the filtering. It is based on the ideals coming from the human perception model. Recent models for human vision adopt the existence of an internal spatial/frequency representation that preserves local and global information [BSI87]. Based on these findings, the image is decomposed into a set of different sub-bands that are convolved images of the input image with a bank of linear and non-linear filters. A quite common filter bank is Gabor filters with different scale and orientation. It has the property that it can segment images having region differences in spatial frequency , density of elements, orientation, phase and energy.

A 2-D Gabor transformation is an oriented complex sinusoidal grating modulated by a 2-D Gaussian function, which is giving by

$$G_{\sigma,F,\theta}(x,y) = g_\sigma(x,y)exp[2\pi jF(x\cos\theta + y\sin\theta)],$$

where

$$g_\sigma(x,y) = \frac{1}{2\pi\sigma^2}exp(-\frac{x^2+y^2}{2\sigma^2}). \tag{2.16}$$

The frequency of the span-limted sinusoidal grating is giving by $F$ and its orientation is specified by $\theta$; $\sigma$ is the scale parameter. The parameter s of a Gabor filter are therefore given by the frequency $F$, the orientation $\theta$, and the scale $\sigma$. The Gabor filter $G_{\sigma,F,\theta}$ gives a complex-valued function, which can be decomposed into real and imaginary parts by

$$G_{\sigma,F,\theta} = G_R + jG_I.$$

Giving an image function $u_0$, defined in a planar domain $\Omega$ which taking real values, the texuture features are captured by the Gabor transform of $u_0$:

$$G(u_0|\sigma,F,\theta) = \sqrt{(G_R * u_0)^2 + (G_I * u_0)^2}.$$

Let $u_0^i, i = 1, \cdots, N$, be $N$ Gabor transforms of the original image $u_0$, obtained for different parameters $(\sigma, F, \theta)$. Then we have a vector-valued image $u = (u_1, u_2, \cdots, u_N)$. We can easily extend the Chan-Vese model to this vector-valued image as follows:

$$F(H(\phi), c_1, c_2) = \mu\Big(\int_\Omega |\nabla H(\phi)|\Big) + \lambda_1 \int_\Omega \sum_{i=1}^{N} |u_0^i - c_1^i|^2 H(\phi) dx + \lambda_2 \int_\Omega \sum_{i=1}^{N} |u_0^i - c_2^i|^2 (1 - H(\phi)) dx.$$

where

$$c_1^i = \frac{\int_\Omega u_0^i H(\phi) dxdy}{\int_\Omega H(\phi) dxdy},$$
$$c_2^i = \frac{\int_\Omega u_0^i (1 - H(\phi)) dxdy}{\int_\Omega (1 - H(\phi)) dxdy}.$$

We can apply the fast algorithm to this vector-valued segmentation model without any difficulty. The experiment results are giving in next section.

## 2.6 Experimental Results

In this section, we present numerical results using our new algorithm on the Chan-Vese segmentation model on various synthetic and real images. All of the calculation use Jacobi iterations. The numerical implementation used is quite simple and only requires a few lines of C++ code. In our experiment, we set $\phi = 1$ inside the level set and -1 outside, $\mu = 0.045$, $\lambda_1 = 1$ and $\lambda_2 = 1$.

First, we use a one dimensional example to illustrate our algorithm. The function is a two phase function which takes value 100 and 20. Although it has noise, therefore theorem 1 does not strictly apply, it still converges in one sweep. See Fig. 2.4.

Fig. 2.5 shows the segmentation results on 2-phase image. The length term is omitted since there is no noise. The image size is $100 \times 100$. We use four different initial conditions, all of them converge to correct solution in one sweep, as compared to more than 200 sweeps in the original method. In fact, it is hard to find an initial condition that it does not work. This also shows that the algorithm is quite robust.

In Fig. 2.6, we show an example that this algorithm does not always work. The initial condition is $\phi = 1$ on the left side and $\phi = -1$ on the right side. Then the averages $c_1$ and $c_2$ will be equal and not satisfy the condition in theorem 1. After one sweep, the part in which $\phi = 1$ will become $\phi = -1$ and vice versa. In the second sweep, it will change back to the initial condition. It will just change back and forth so it does not converge. If we use a Gauss-Seidel iteration instead of Jacobi iteration, then it will converge in one sweep.

In Fig. 2.7, we apply our algorithm to an image with noise. In this example, the length term in the energy expression is considered. Convergence is achieved in less than 5 sweeps, as compared to more than 400 sweeps in the original method. We show the result at sweep 1, 2, 3 and 4. We note that there is no special handling needed to detect the interior contours.

In Fig. 2.8, we show different initial conditions (a), (b), (c) and plot the energy versus iterations. The length term is considered. Note how fast our algorithm is. All of them converge in less then 8 sweeps. They also converge to the same result (d).

Fig. 2.9 shows the results of the Chan-Vese model without the length term. It converges in 4 sweeps. Note that it does not converge in exactly one sweep because it is not a 2-phase piecewise constant image. This result is very close to the result we want and we can either do a few more steps of denoising or use a

PDE-based algorithm to reach the final result.

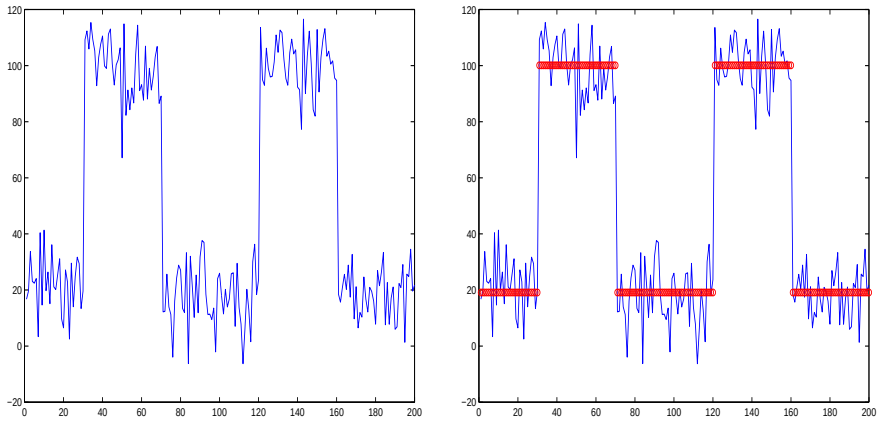
In Fig. 2.10, we show the result of our new algorithm followed by level set evolution. We first use 4 sweeps of new algorithm and then use 5 iteration of original Chan-Vese method. We can also use other methods such as denoising or diffusion to remove the noise.

Fig. 2.11 shows the segmentation result of a noise image. In this example, we do not consider the length term. Although noise is presented, we still get correct result after 6 sweeps. The image size is $240 \times 240$.

Fig. 2.12 shows the segmentation result for a 3-phases image. We use two level sets to represent the image. (a) is the initial contours overlay on original image, (b) is the segmentation results. The image is 64x64 and converges in 1 sweep. Since the energy which is minimized is not convex, and also there is no uniqueness for the minimizers, the algorithm may not converge. We also have no theoretical result like theorem (1) for multiphase images.

Fig. 2.13 shows an example of segmenting a shading image. (a) is the original image which consists of two linear parts. (b) is the segmentation result of piecewise constant approximation. (c) is the segmentation result of piecewise linear approximation, it needs only 6 sweeps to converge. We can see that it converges to correct segmentation.

Fig. 2.14 shows the texture segmentation. (a) is the original image, (b), (c), (d) are the Gabor transformation at different scale and direction, and (e) is the segmentation result.



Original 1-D Image Segmented Image

Figure 2.4: 1-D Example. No length term in this example, converges in one sweep.

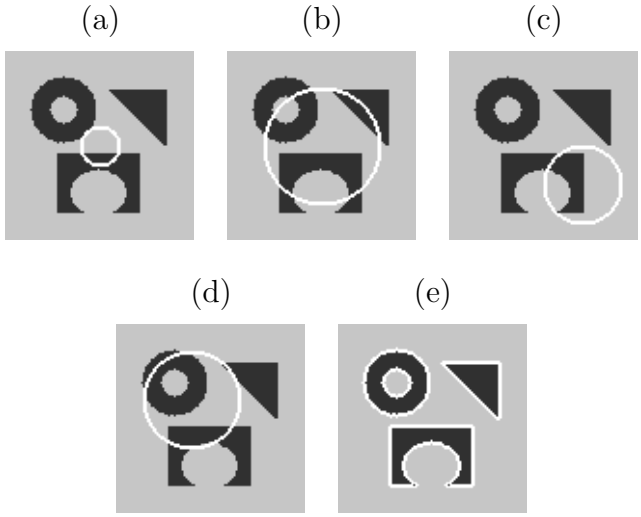



Figure 2.5: Segmentation of 2-phase image. (a),(b), (c), (d) are four different initial conditions, all of them have the same result after 1 sweep. The result is in (e). The image size is $100 \times 100$. Note that interior contour of the circle is automatically detected.

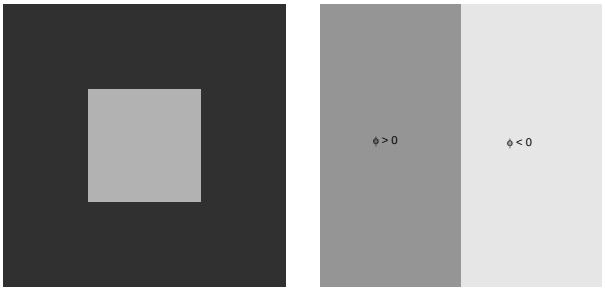



Figure 2.6: An example to show that the algorithm does not always work. Left: image we want to segment. Right: the initial $\phi$. $\phi = 1$ on left side and -1 right side. The average on left side and right side are equal, so the necessary condition of theorem 1 does not apply.

## 2.7 Concluding Remarks

We propose a fast algorithm for a special kind of optimization problems which can be formulated by level set. Instead of solving the corresponding Euler-Lagrange equation, We compute the energy directly and see how the energy will change when we change the sign of a point. The main advantage is that the gradient of the functional is not needed. Thus, it can be applied to more broad range of optimization problems. This algorithm is very successful when we apply it to the Chan-Vese model. We proved that for a 2-phase image, it converges in one sweep with most initial condition. The method can be easily extended to higher dimensional problems such as 3-D segmentation and clustering.

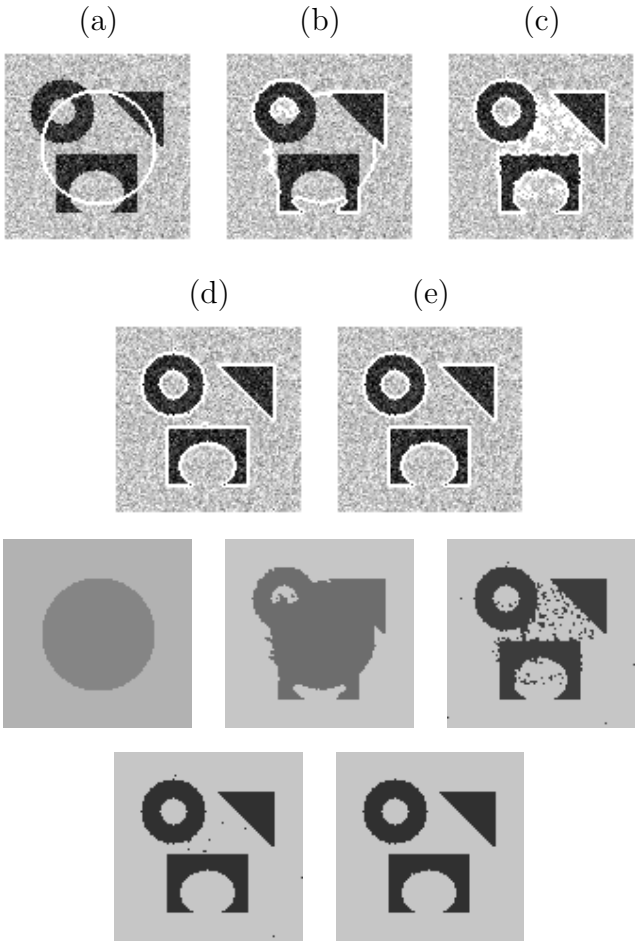



Figure 2.7: Detection of different objects from a very noise image. Top: $u_0$ and the contour. (a) initial $\phi$, (b) after 1 sweep. (c) after 2 sweeps, (d) after 3 sweeps, (e) after 4 sweeps. Bottom: the piecewise constant approximation of $u_0$. Even though theorem 1 does not strictly apply, the algorithm still converges very fast.

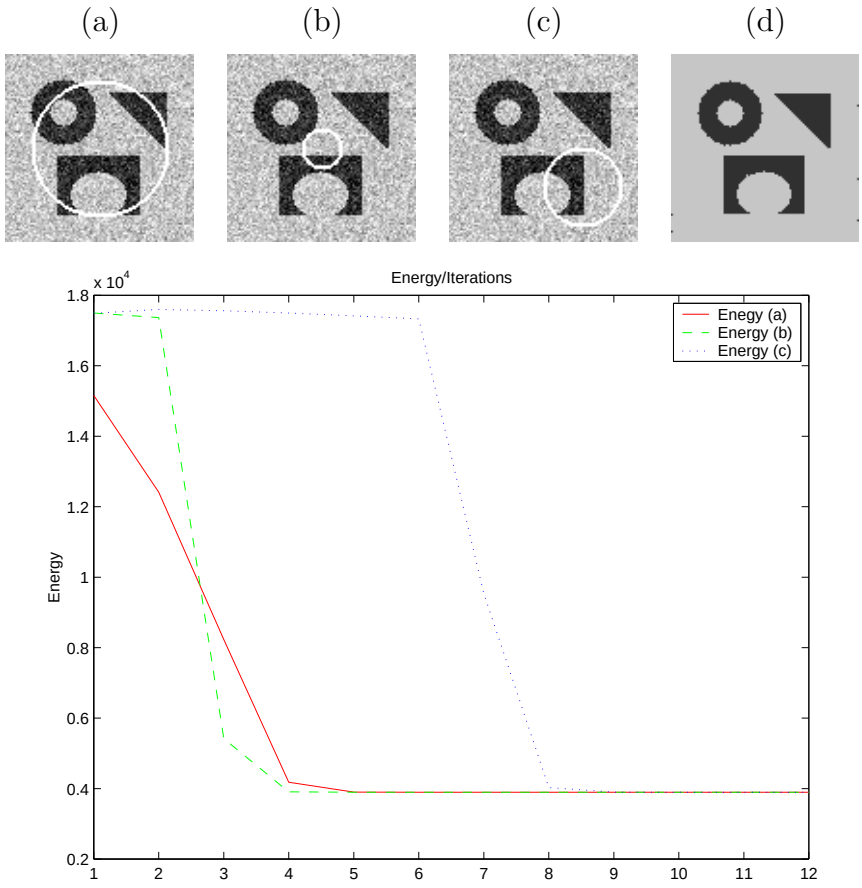



Figure 2.8: Three different initial conditions and the corresponding energy versus iterations. In this example, the length term is included in the objective functional. Note that all 3 initial conditions converge to the same result.

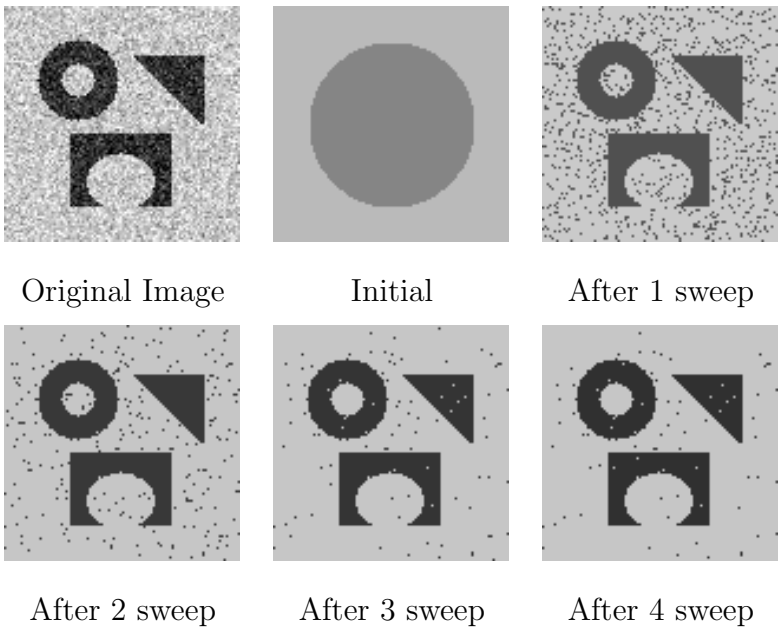



Figure 2.9: Chan-Vese model without length term. Note that even though theorem 1 does not strictly apply, the algorithm still converges in only 4 iterations.

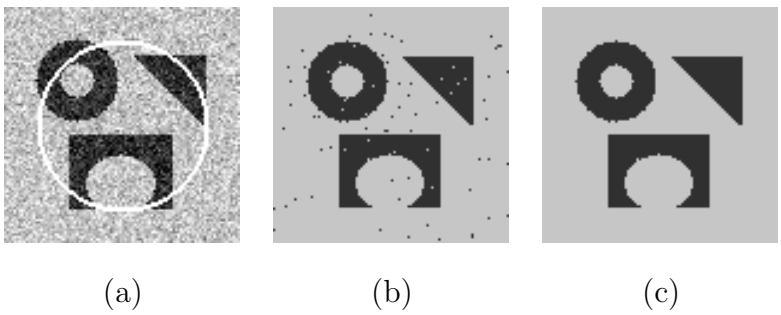



Figure 2.10: Combining optimization and PDE evolution. (a) Initialization, (b) Result of fast algorithm using 4 sweeps, note that there are still some pixels which have not converge and show as "noise". (c) Using 5 sweeps PDE evolution starting from (b). Note that all the "noise" have converged.

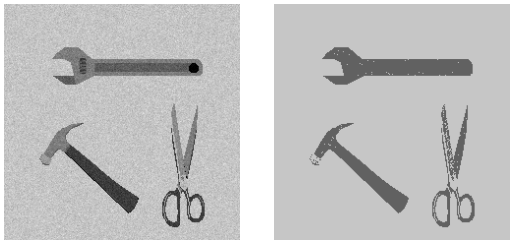


Figure 2.11: Segmentation of a noisy image. (a) Original image, (b) Result of piecewise constant approximation. The size of image is 240 x 240 and it converges in 6 sweeps. In this example, we did not consider the length term.

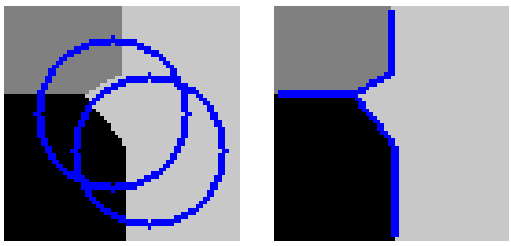
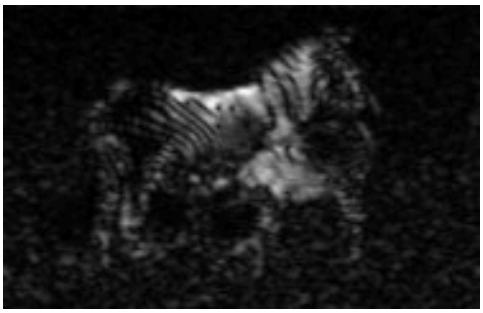


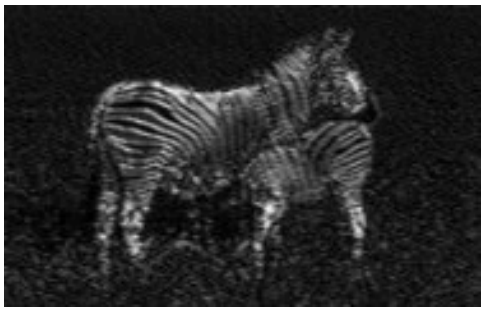
Figure 2.12: Segmentation of 3-phase image. (a) is the initial contours overlay on original image, (b) is the segmentation results. The image is 64x64 and converges in 1 sweep.

(a) (b) (c)

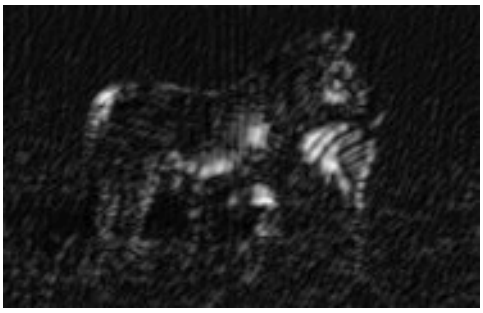
Figure 2.13: Piecewise linear approximation to image. (a) Original image, (b) Result of piecewise constant approximation, converges in 4 sweeps. (c) Result of piecewise linear approximation, converges in 6 sweeps.

(a)





(b) (c)



(d) (e)

Figure 2.14: Texture segmentation. (a) original image, (b), (c), (d) the Gabor transformation at different scale and direction, (e) segmentation result.

# CHAPTER 3

# Inpainting

## 3.1 Introduction

Texture synthesis or inpainting refer to select a texture and synthesis it inside the region to be filled-in. It is important for many applications in computer graphics, vision, and image processing.

The problem we address in this chapter is texture synthesis constrained by boundary matching with a seed texture. Given an image $u_0$ where a few pixels or a whole region $\Omega$ is missing, we wish to recover the lacking information in the smoothest possible way to the eye. A few principles come to mind when trying to give this problem a more precise statement :

1. The seed image should provide a guideline to the synthesis of the missing pixels. The result of inpainting should locally be visually close to parts of the known image.

2. In addition to considering the seed image, one might wish to consider a set of images, build a summarizing account of their properties, and use this as learned *a posteriori* knowledge.

3. Our *a priori* knowledge of some typical image properties (there are uniform regions, edges, texture, etc.) should provide other indications for the inpainting.

4. At yet a higher level, our understanding of the scene can give us hints at how to inpaint properly.

Numerous approaches have been proposed for texture synthesis and analysis. Zhu et. al [ZWM98] model texture as a Markov Random Field and use Gibbs sampling for synthesis. However, Gibbs sampling is computationally expensive and very slow. Heeger and Bergen [HB95] model textures by matching marginal histograms of image pyramids. While this technique works well for highly stochastic textures, the histogram is not powerful enough to represent more structured texture patterns. DeBonet [DeB97] synthesizes new images by randomizing an input texture sample while preserving the cross-scale dependencies. This method works better than [HB95] on structured textures, but it can produce boundary artifacts if the input texture is not tileable. Simoncelli and Portilla [SP98] generate textures by matching the joint statistics of the image pyramids. Their method can successfully capture global textural structures but fails to preserve local patterns.

Efros and Leung [EL99] proposed a non-parametric method for texture synthesis. The texture synthesis process grows a new image outward from an initial seed, one pixel at a time. They model texture as a Markov Random Field (MRF), and the conditional distribution of a pixel giving all its neighbors synthesized so far is estimated by querying the sample image and finding all similar neighborhoods. This method works well for a wide variety of synthetic and real-world textures. Wei and Levoy [WL00] uses a synthesis approach which, although based on the Markov Random Field model of texture, avoids explicit probability function construction and consequent sampling from it and generate the texture through a deterministic search process. This is accomplished by generating the output image pixel by pixel in scanline order, choosing at each step a pixel from

the sampling image which neighborhood is most similar with respect to a specified measure to the current available neighborhood in the texture being synthesized.

The algorithm we describe in the next section, in the form introduced in [WL00], is at a very 'low level' since it only concentrates on the first approach[1]. It was originally designed for texture synthesis in a broad sense and it is fair to say that its application to inpainting tasks is not a significant idea on its own. Variational or PDE-based inpainting methods [BBC01, CS02], on the other hand, are based on some a priori information (e.g. the functional to minimize) as well as data fitting (e.g. the boundary condition), so they fall in between the third and first categories. Markov random field texture models, cfr. [ZWM98], consist in building a big probability distribution from a large set of images, and then sampling it in order to synthesize the learned patterns. This follows approach number 2 above.

Section 3.2 reviews well-known algorithms for texture synthesis, as introduced in [EL99, WL00]. The core of the paper, in sections 3.3 and 3.4, attempts at providing a framework to understand these algorithms. Section 3.5 discusses some experiments and directions for future research are given in section 3.6.

## 3.2 The Algorithm

We are given a known image (intensity function) $u_0(\alpha)$ defined on the pixels $\alpha \in I \backslash \Omega$ for some missing region $\Omega$. The strategy for recovering the unknown intensities $u(\alpha)$ for $\alpha \in \Omega$ is to copy them as $u(\alpha) = u_0(F(\alpha))$ using a correspondence map $F : \Omega \to I \backslash \Omega$. The following steps, depicted in Figure 3.1, build $F$.

[1]This is why we will not use the word 'model' to refer to the various ways of formalizing this algorithm. There is not much to model here.

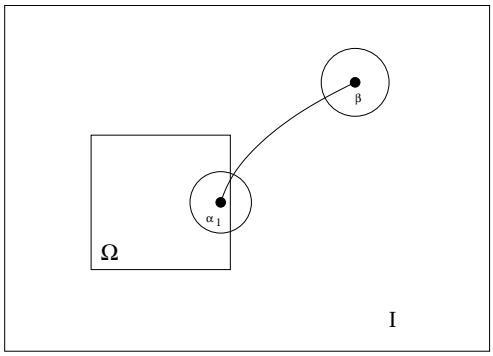



Figure 3.1: The correspondence map $F : \Omega \to I \backslash \Omega$

- Pick a pixel $\alpha_1 \in \Omega$, preferably near the border, and consider a small neighborhood of $\alpha_1$ made of known pixels $\beta \in I \backslash \Omega$.
- For every pixel in $I \backslash \Omega$ consider its corresponding neighborhood.
- Select the pixel $\beta \in I \backslash \Omega$ which best matches $\alpha_1$ in terms of neighborhood distance. Closeness can for example be measured using the distance

$$d_N^2(\alpha, \beta) = \sum_{\gamma \in N} |u_0(\alpha - \gamma) - u_0(\beta - \gamma)|^2. \tag{3.1}$$

  $N$ is that small neighborhood around the origin. We have made the assumption that the neighborhoods of both $\alpha$ and $\beta$ is fully contained in the known region $I \backslash \Omega$. Then take $F(\alpha) = \beta$.
- Repeat the same procedure for the next pixel $\alpha_2$, considering that $\alpha_1$ is now part of the known image. The synthesis procedure is complete after just one sweep of $\Omega$.

This basic scheme can be made more efficient in a few different ways. One can restrict the search for a good neighborhood match to a subset of the seed pixels, e.g. chosen randomly and not too far from the current $\alpha$. If we call the 'codebook' the set of all neighborhood candidates, this is actually codebook

pruning. A good data structure on this codebook can also speed up the search (by making it approximate). The construction of the correspondence map can also be done in a multiresolution way, by successive refinements on embedded grids. The extension to color images is obvious if we understand that $u$ and $u_0$ are vector-valued, and that the absolute value $|\cdot|$ becomes the $l^2$-norm. We encourage the reader interested in implementation issues to refer to [EL99, WL00].

## 3.3 The Analysis of the Algorithm

Let us go back to the basic scheme for the sake of its analysis. As we have seen each step consists in solving the following problem,

$$\min_{F(\alpha_i)} d_N(\alpha_i, F(\alpha_i))$$
$$\text{given } F(\alpha_1), F(\alpha_2), \ldots, F(\alpha_{i-1}).$$

This can also be seen as the maximization of the conditional probability

$$P(F(\alpha_i)|F(\alpha_1), F(\alpha_2), \ldots, F(\alpha_{i-1})) = e^{-d_N^2(\alpha_i, F(\alpha_i))} \tag{3.2}$$

Of course $u_0$ is hidden in the condition as well, we drop it for notational convenience. That the values of $F(\alpha_{i+1}), \ldots$ are not present in the condition is what is sometimes called the 'causality' property of the neighborhood. This is a clever way to get the corresponding joint probability to factorize,

$$\begin{aligned} P(F(\alpha_1), \ldots, F(\alpha_n)) =& P(F(\alpha_1))P(F(\alpha_2)|F(\alpha_1)) \ldots \\ & P(F(\alpha_n)|F(\alpha_1), F(\alpha_2), \ldots, F(\alpha_{n-1})). \end{aligned} \tag{3.3}$$

Using the correspondence backwards, we can define the global 'inpainting energy' as

$$E(F) = -\log P(F(\alpha_1), \ldots, F(\alpha_n)) = \sum_{\alpha_i \in \Omega} d_N^2(\alpha_i, F(\alpha_i)). \tag{3.4}$$

It would be particularly nice to obtain a correspondence map $F$ as the absolute minimizer of this indicator since it would mean that *we would have pasted information into the masked region to make it locally look as much as possible like the seed region*, in the sense defined in the expression.

Obtaining the global minimum of (3.4) is a different and much more difficult problem than performing the successive neighborhood matchings (3.3). When designing a method to solve such a hard optimization problem there is a trade-off to solve between the ability to make the energy decrease and the ability to escape from local minima. The successive local minimizations can be considered as a rather naive but very fast greedy procedure to reach a good local minimum. This is how we understand the success of the algorithms in [EL99, WL00]. Note that given an arbitrary initial $F$ the energy (3.4) is not guaranteed to decrease at every iteration.

The framework is now set to consider non-causal neighborhoods, thereby restoring rotational invariance at the level of the objective to be minimized. The distance $d_N(\alpha, F(\alpha))$ can be built using a neighborhood $N$ symmetric around the pixel $\alpha$. The natural extension of the basic algorithm is to start from an arbitrary $F$ and refine it by multiple sweeps of the original procedure over the inpainting domain $\Omega$. A good way to achieve approximate rotation invariance at the level of the algorithm is to give up the raster scan ordering of the pixels and rather scan from the border of $\Omega$ to its interior. Note that for a noncausal neighborhood the joint probability $e^{-E(F)}$ no longer neatly factorizes as above and it would actually be forbidden to consider $e^{-d_N^2(\alpha,F(\alpha))}$ as a conditional probability.

Instead let us start from the total probability distribution and see how we can decompose it using Bayes' rule. The discussion below applies equally to both causal and noncausal neighborhoods. For every pixel $\alpha_i \in \Omega$ define $\alpha_i^c$ as its

complement in $\Omega$, and consider the conditional probabilities

$$P(F(\alpha_i)|F(\ \alpha_i^c)) = \frac{P(F(\alpha_1),\ldots,F(\alpha_n))}{P(F(\ \alpha_i^c))}.$$

Their expression can be found by isolating the dependence on $\alpha_i$ in the expression of the total energy : $P(F(\alpha_i)|F(\ \alpha_i^c)) = Ce^{-\tilde{d}_N^2(\alpha_i,F(\alpha_i))}$, where ($N$ is a neighborhood around the origin)

$$\begin{aligned}\tilde{d}_N^2(\alpha,F(\alpha)) = &\sum_{\gamma\in N}|u_0(\alpha-\gamma)-u_0(F(\alpha)-\gamma)|^2 \\ &+\sum_{\gamma\in N}|u_0(F(\alpha))-u_0(F(\alpha+\gamma)-\gamma)|^2.\end{aligned} \tag{3.5}$$

It is instructive to compare this expression with equation (3.1). The first term is exactly $d_N^2$ and measures neighborhood closeness at $\alpha$ vs. $F(\alpha)$. The second term is present because $\alpha$ is itself a neighbor of the pixels which are in its neighborhood. It measures how much changing $F$ at $\alpha$ disrupts the neighborhood matches for the pixels surrounding $\alpha$. Of course the basic algorithm can be modified by using this new distance instead of $d_N$. It has the advantage of making the global energy decrease at every step, and is therefore guaranteed to reach a local minimum. However this idea comes with a big drawback. The algorithm becomes a pure descent and loses all its 'creativity' to find a good local minimum. Typically it would produce uniformly gray or colored regions by copy-pasting pixels from a flat smooth region in the seed image. This again emphasizes the true problem here : the need for clever algorithms to minimize our functional.

If we model the image $u$ as a function from $I \subset \mathbb{R}^2$ to $\mathbb{R}$, and the correspondence map as a function $F$ from $\Omega \subset R^2$ to $I\backslash\Omega \subset \mathbb{R}^2$, the total inpainting energy

to be minimized over $F$ is expressed as[2]

$$E(F) = \int_{\Omega} dy \int_{\mathbb{R}^2} dx \chi(x) |u_0(F(y-x)) - u_0(F(y)-x)|^2, \tag{3.6}$$

where $\chi(x)$ is a bump function chosen to indicate the neighborhood of 0. Then, as before, $u(x) = u_0(F(x))$ for every $x \in \Omega$. This 'energy functional' is very different from, say, the TV functional for inpainting. Here the minimization is not made over $u$, taking on values in some intensity space, but over the intermediate $F$, taking on values as coordinates of points in the image $I$ itself. A solution $F$ can be highly nonsmooth. As we said before, the energy $E(F)$ is far from having convexity properties.

It is also at a much more elementary level than a regularization functional, in the sense that it is defined with poor learned (a posteriori) knowledge or subjective (a priori) assumption on what the solution should look like. Minimizing $E(F)$ is not a model, it is hardly more than one possible low-level formulation of the inpainting problem. A pruned codebook, made of pixel neighborhoods and built from a single texture or image sample, does not qualify satisfactorily as a set of model parameters. Of course the above comments do not question the efficiency of the algorithms for practical purposes.

## 3.4 Deterministic vs. Probabilistic

So far we have associated probability and energy, through $E = -\log p$ or $p = e^{-E}$, but this identification is only formal. The models people build from these two concepts can be very different.

[2]In this formulation we discard all sorts of problems related to the treatment of boundaries. For example the neighborhood of $F(y)$ has a possibly nonempty intersection with $\Omega$, and we could account for that by restricting the range of $F$ or by writing $F(F(y) - x)$ instead of $F(y) - x$. If necessary the map $F$ is taken to be the identity on $I \backslash \Omega$.

- An 'energy' functional $E(F)$ should be *minimized*. For our purposes it is a criterion that sorts every possible argument $F$ in a definite order, from the best one (lowest energy) to the worst ones (highest energy). We are therefore interested in the (often unique) argument of the minimum. Examples related to our purpose include regressions in statistics, denoising-deblurring problems addressed by variational or Bayesian methods in image processing, and of course variational inpainting [CS02].

- A probability distribution $p(F)$ should be *sampled*. For our purposes it is a criterion that sorts each event into categories, e.g. as typical or non-typical. All the typical events have approximately the same probability. All the non-typical events have approximately zero probability, or have a very high probability but then are by far outnumbered by the typical events. We are interested in any one of these typical events. Examples include the simulation of Markov Random Field models in statistical mechanics or texture synthesis [Br99, ZWM98].

In some sense a probability model is 'weaker' than an energy model. It does not manage to rank all the events into a significant one-dimensional scale. Moreover, the most probable outcomes of a random vector need not look like the vast majority of typical outcomes, so it might turn out to be a bad idea to try and maximize the probability. The classical example is a multivariate random variable made of i.i.d. $N(0,1)$, which typical samples look like 'noise', but which most probable outcome is the vector that is identically equal to zero (this is not 'typical').

The two paradigms are usually used in very different contexts to address very different questions. However it is not clear yet where solutions to inpainting should belong. As far as the simple 'neighborhood-matching' algorithms are

concerned, most authors seem to classify them as sampling from a distribution. A hint in this direction is the complete factorization of the probability distribution (3.3), in the case of a causal neighborhood, which results is a straightforward sampling strategy : sample the successive conditional distributions one after the other. This is for example the core of the approach in [PP93].

We rather believe that the algorithms of section 2 should be viewed as descent procedures to minimize the energy functional previously introduced. As far as we experienced, no harm was done when trying to reach a good minimum. The descent never seems to reach atypical highly probable states. Thus no need for a careful sampling. The pixel-pasting-by-neighborhood-matching approach is presumably not involved enough to require a true probabilistic modeling. The numerical experiments supporting this conclusion are shown in the next section. We do not rule out the possibility that better algorithms minimizing the energy functional more faithfully might disprove the claim.

This provides an explanation for the difference of performance of the algorithms in [EL99] and [WL00] : the former approach insists on sampling the conditional probabilities whereas the latter approach simply maximizes them. This results in a much faster algorithm without loss of visual quality of the synthesized texture. The numerical experiments supporting this conclusion are shown in the next section.

There is a way to re-introduce probability in our setting, not at a modeling level but rather as a tool to solve the optimization problem. When getting trapped at a nonsatisfactory local minima is the inevitable faith of naive descent procedures, people often resort to 'stochastic descents' to have a better chance of reaching the global minimum. For instance, simulated annealing intuitively keeps the system from freezing at a high-energy state by "shaking it hard enough" dur-

ing a slow cooling. For the particular case of our pixel synthesis algorithm, it could mean randomly assigning a disadvantageous target $F(\alpha)$ to the current pixel $\alpha$. Of course such a procedure would considerably slow down the descent but could prove useful to output a better looking result. One reference for these probabilistic optimization algorithms is [Br99]. These tools turn out to be closely related to sampling strategies, but our message here is that the deterministic vs. probabilistic nature of the model investigated is a different question. It should be answered prior to choosing an algorithm.

## 3.5 Numerical Experiments

We run the algorithm as described in section 2, with minor improvements.

- Inpainting is done at a very coarse scale first, with a random map $F$ as initialization. Using an appropriate interpolation procedure one can then recursively refine this guess at finer and finer resolutions. This way interactions can occur at a much larger scale than the size of the neighborhood (a 3 by 3 square, say). See [WL00].

- The filling-in of the missing pixels is done from the border of the missing region to the inside in order to respect rotational invariance. This procedure is repeated as many times as necessary. This is what we refer to as the number of 'sweeps'.

- An exhaustive search is far from being necessary when looking for a good match among the pixels in the seed image. Randomly choosing a subset of these pixels, of size typically smaller by a factor of 10, already suffices to produce visually similar results.

- There are a few natural pixels to include in the subset mentioned above ; they should not all be chosen randomly. Suppose all the neighbors of a pixel $\alpha$ have already been inpainted, then a good candidate for $F(\alpha)$ would be $F(\alpha - \beta) + \beta$ for $\beta$ close to the origin. This is described in [Ash01].

Fig. 3.2 is a synthetic geometric example. On the left is the image to be inpainted, the 'noise' indicating the mask where the pixel intensity values are occluded or missing. On the right is the result after inpainting. In this example, the minimum of the inpainting energy is zero. Fig. 3.3 is de Bonet's texture sample nr. 161. Again, (a) is the occluded image, (b) is the inpainting result and (c) is the energy vs. the number of sweeps. The energy decreases quickly in a few sweeps and then stagnates till some approximately steady state is reached. Note that plot (c) was obtained only from running the algorithm at the finest resolution. Fig. 3.4 shows that the algorithm can also be very successfully applied to textured images. Fig. 3.6 is the 'Barbara' image that contains both a texture and a cartoon part : (a) is the degraded image, (b) is the result of inpainting by correspondence map. We observe that the texture part is often well recovered. (c) is the result of TV inpainting. We can see that the texture part is not recovered at all. Another very efficient way to process this image would be to decompose the image as a texture plus cartoon image and then use a different inpainting method for each part. See [BVS02] for details.

In all the above examples and many others the quality of the result does not degrade as the number of sweeps gets very large. This supports the conclusion that the model behind the algorithm is deterministic and not probabilistic, in the sense discussed above.

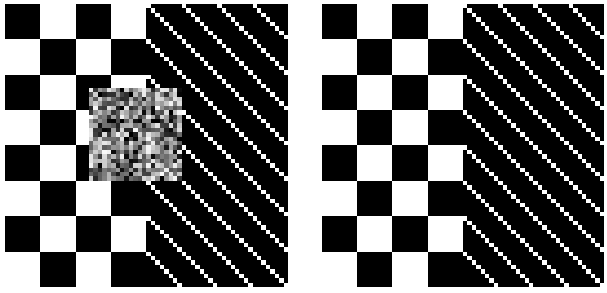


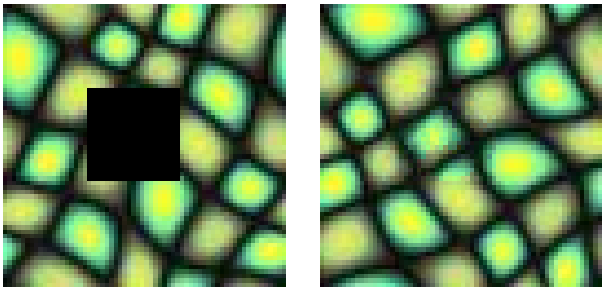
Figure 3.2: Inpainting a synthetic image. Left: The image to be inpainted. The 'noisy square' in the middle indicates the occluding mask. Right: After inpainting.

Another interesting experiment to test this claim is to start with the *occluded original image* as initial guess, and apply the algorithm[3]. We observe that, on toy inpainting problems such as the one in Fig. 3.7, the 'steady-state' is visually close to the original image and the inpainting energy is not significantly lowered. Once again, this validates the claim that minimizing an energy is the right framework behind the algorithm. If the algorithm had degraded the image substantially, we could have resorted to sampling strategies to avoid that phenomenon ; but this is not the case here. Note that running the algorithm with the 'forbidden' occluded part of the image as initial data could be used in practice to erase unwanted information.

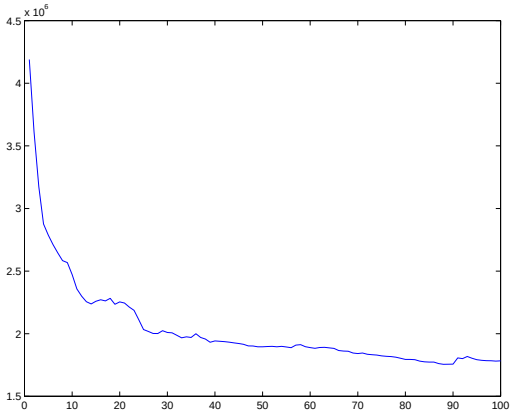
## 3.6 Extensions

The low-level inpainting solution consisting of minimizing (3.6) will probably gain from being formulated using some additional a priori information. A few paths can be followed.

[3]No multiresolution strategy is adopted here. The initial inpainting map is computed in an obvious way from the original image by neighborhood matching.

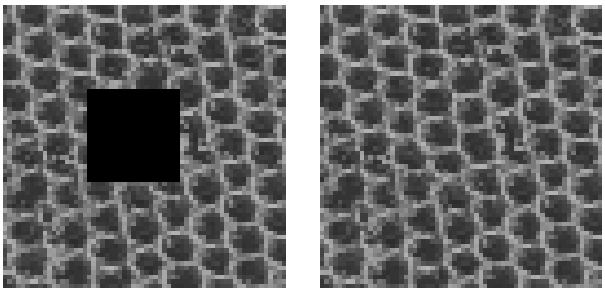


(a) (b)




(c)

Figure 3.3: Inpainting a textured image. (a) The occluded image. (b) After inpainting (c) Inpainting energy vs. number of iterations.



(a) (b)

Figure 3.4: Inpainting a textured image. (a) The image to be inpainted. (b) After inpainting.

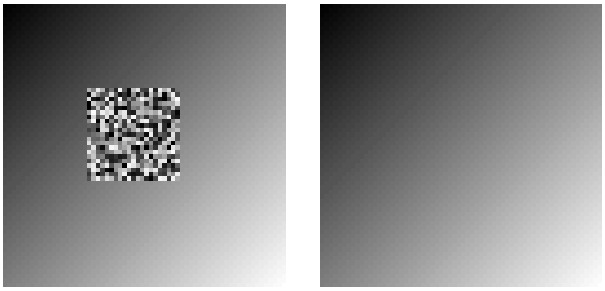


Figure 3.5: Inpainting a linear smooth image. Left: the image to be inpainted. Right: After inpainting. Although this image is not just texture, the algorithm can still be applied.

- There is a notion of visual closeness associated with the energy (3.6). Namely, for every synthesized pixel there exists a pixel in the seed image in good agreement in terms of similarity of the neighborhoods. It would however make little sense if the correspondence map took the observer to a very distant pixel everytime a step is taken in the missing region. Rather than isolated pixels, we expect reasonably large patches to be pasted into that region. This amounts to requiring 'smoothness' of the correspondence map and suggests adding the following penalization term,

$$\begin{aligned} \min \tilde{E}(F) = &\int_{\Omega} dy \int_{\mathbb{R}^2} dx \chi(x) |u_0(F(y-x)) - u_0(F(y)-x)|^2 \\ &+ \lambda \int_{\Omega} dy \, \|\nabla F(y) - I\|_F, \\ &u(y) = u_0(F(y)). \end{aligned}$$

  In other words the correspondence map should locally look like the identity ($\|\cdot\|_F$ is the Frobenius norm). The parameter $\lambda$ weights the importance of each term. How to efficiently implement the minimization of this new energy is the interesting problem. A step in this direction is [Ash01] where the author does not just select the pixel candidates randomly in the seed image but also according to what has already been synthesized. Preference

(a)

(b)

(c)

Figure 3.6: Inpainting a real-life image. (a)Image with missing information, (b) Result of inpainting by correspondence map. We can see that the texture part is well recovered. (c) Result of TV inpainting.

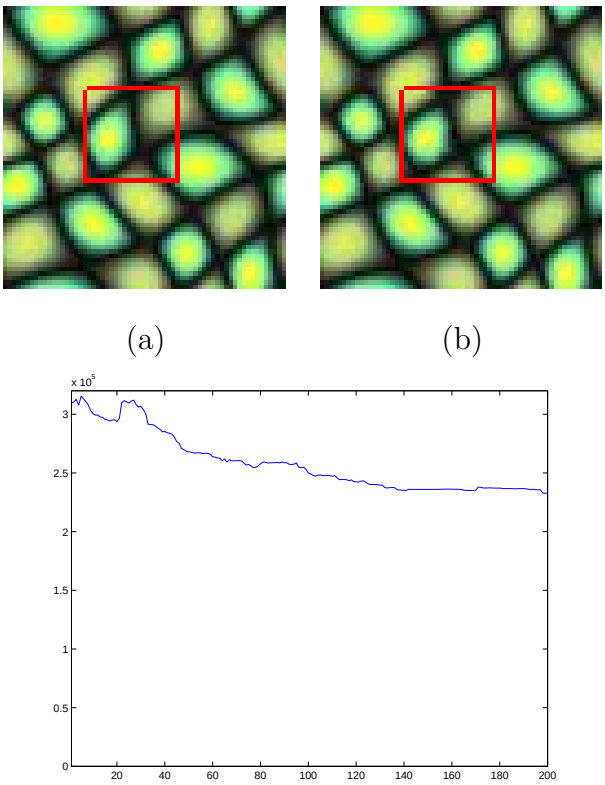



Figure 3.7: Taking original occluded image as initial guess and applying inpainting. (a) Original image. (b) After inpainting. (c) Inpainting energy vs. number of iterations. Note that the energy does not always decrease. *This example is not supposed to show the performance of the method, see explanations in the text.*

is precisely given to these pixels that extend the correspondence map so as to copy larger patches into the missing region. See also comments in section 5.

- Naturally, the algorithm is well-suited for texture synthesis but often fails on reproducing geometrical features of the image. This is precisely what TV inpainting does reasonably well for you. It is therefore tempting to write combined models such as the following minimization problem.

$$\min_{u,F} E(F) + \lambda_1 TV(u) + \lambda_2 ||u_0(F(x)) - u(x)||_2^2,$$

where $TV(u)$ is the TV norm of $u$, $E(F)$ is the total inpainting energy defined in (3.6). The issue would then again be to find a clever algorithm to minimize this. It is probably a good idea to make $\lambda_1$ decrease as the number of iterations (sweeps) increases. This situation would more or less correspond to choosing the TV inpainting as initial guess for the usual algorithm.

# CHAPTER 4

# Adaptive Total Variation Denoising

## 4.1 Introduction

Random noise is unavoidable in many phases of manipulating images. This random noise makes it difficult to do further image processing such as edge detection, pattern recognition, and object tracking, etc.

The classical algorithm for image denoising is based on least squares estimate. This $L^2$ norm based algorithm amounts to solving a heat equation. The main advantage of this method is that it is linear and stable, thus can be easily implemented. But it suffers seriously from the fact that it can not keep the jumps in 1-D signals and edges in 2-D images. Denoising model based on the least squares estimate will inevitably blur the sharp edges.

To overcome this difficulty, Rudin, Osher and Fetami [ROF92] proposed a technique based on the minimization of the Total Variation (TV) norm subject to some noise constraints. This model gives simple but efficient measurement for image debluring and denoising. The main benefit of TV model is that it does not penalize discontinuity in the image, thus keep the edges very well, but the corresponding Euler-Lagrange equation is not trivial to compute since it is highly nonlinear and not well-posed in strong sense. Furthermore, this method suffers from the *staircase effect*, i.e. transform smooth regions(ramp) to piecewise constant regions(stairs).

In this chapter, we present an algorithm based on an adaptive $L^{1+\epsilon}$ norm for $0 < \epsilon < 1$, its boundary case when $\epsilon = 0$ is nothing but the TV norm, therefore such method so called by us an adaptive Total Variation based denoising. We will show that, with this small input, the resulting Euler-Lagrangian equation is well-posed and its numerical implementation is convergent and stable. It is also shown that such adaptive method can reduce the staircase effect effectively.

We note that the same method was used in [PM97] for numerical test, but the authors did not pursue it there.

This chapter is organized as follows: In section 2 we briefly review the TV model and its implementation methods. In section 3 we introduce our new model. Numerical implementation and stability properties are given in section 4. In section 5, we show experimental results, and we end the chapter by a brief concluding section.

## 4.2 Total Variation Denoising

In recent years, much work in image restoration has been done using total variation and the results are very promising. The restored image is the solution of:

$$\begin{aligned} &\min_u \int_\Omega |\nabla u| dxdy, \\ &\text{subject to} \quad \frac{1}{2}\int (u - u_0)^2 dxdy = \sigma^2. \end{aligned} \tag{4.1}$$

To solve this minimization problem, we usually solve its Euler-Lagrange equa-

tion

$$-\nabla \cdot \left( \frac{\nabla u}{|\nabla u|} \right) + \lambda(u - u_0) = 0, \quad \text{in } \Omega, \tag{4.2}$$

$$\frac{\partial u}{\partial n} = 0, \quad \text{on } \partial\Omega. \tag{4.3}$$

Note that the equation (4.2) is not well defined at points where $\nabla u = 0$, due to the presence of the term $1/|\nabla u|$, it is common to use a slightly perturbed version of TV norm:

$$\int_\Omega \sqrt{|\nabla u|^2 + \beta}\, dxdy, \tag{4.4}$$

where $\beta$ is a small positive number. It is shown in [AV94] that the solutions of the perturbed problems converge to the solution of (4.2) when $\beta \to 0$.

Equation (4.2) can be seen as the limit of the following problem:

$$u_t = \Delta_p u, \tag{4.5}$$

where p-Laplacian is defined as $\Delta_p = div(|\nabla u|^{p-2}\nabla u)$. This problem is well-posed in $W^{1,p}(R^2)$ for $p \in (1, \infty)$. The p-Laplacian is degenerate in critical points of $u$. In regular points of $u$ the vector $n(x) = -\nabla u(x)/|\nabla u(x)|$ is well defined, and one can write

$$\Delta_p u = |\nabla u(x)|^{p-2}((p-1)u_{nn} + (N-1)H(x)u_n),$$

where $H(x)$ is the mean curvature of the level surface $\{y|u(y) = u(x)\}$ passing through $x$, and the subscript $n$ denotes differentiation in normal direction $n$. For $p \to 1$ this operator becomes degenerate across level surfaces of $u$ even in regular points.

In practice, there are two ways to solve (4.1), the first one is to solve the Euler-Lagrange equation directly. The second is using gradient decent method.

The basic idea of direct methods is to linearize the Euler-Lagrange equation and solve it. One of the difficulties of solving this equation is its highly nonlinear and non-differential term. Vogel and Omen [VO96] proposed the following fixed point iteration to solve the equation (4.2):

$$-\nabla \cdot \left( \frac{\nabla u^{k+1}}{|\nabla u^k|} \right) + \lambda(u^{k+1} - u_0) = 0. \tag{4.6}$$

This equation is robust but linear convergent.

Chan, Golub and Mulet [TM99] used interior-point primal-dual implicit method to solve (4.2). The main idea is to introduce

$$w = \frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}$$

as a new variable and replace (4.2) by the following system of nonlinear partial differential equations

$$\begin{cases} -\nabla \cdot \omega + \lambda(u - u_0) = 0, \\ \omega\sqrt{|\nabla u|^2 + \beta} - \nabla u = 0. \end{cases}$$

Osher, Rudin and Fetami [ROF92] used a parabolic equation with time as a evolution parameter, or equivalently, the gradient descent method

$$\begin{cases} u_t & = \nabla \cdot (\frac{\nabla u}{|\nabla u|}) - \lambda(u - u_0), \quad \text{for } t > 0, (x, y) \in \Omega, \\ \frac{\partial u}{\partial n} & = 0, \quad \text{on } \partial\Omega. \end{cases} \tag{4.7}$$

However, this evolution procedure converges very slowly to its steady state since the parabolic term is singular for small gradients. In fact, the stability constraint(CFL condition) of this equation is:

$$\frac{\Delta t}{\Delta x^2} \leq c|\nabla u|$$

The equation 4.7 is closely related to anisotropic diffusion and mean curvature motion [MS95]. In fact, we can think this equation as the level-set equation with

normal speed $(\nabla \cdot \frac{\nabla u}{|\nabla u|} - \lambda(u - u_0)/|\nabla u|$. In order to avoid the restriction of the CFL condition, Marquina and Osher [MO00] proposed a new time dependent model that accelerates the movement of level curves of $u$ and regularizes the parabolic term in a nonlinear way. The evolution equation is:

$$u_t = |\nabla u| \nabla \cdot \left( \frac{\nabla u}{|\nabla u|} \right) - |\nabla u| \lambda (u - u_0).$$

The CFL condition for this equation is $\frac{\Delta t}{\Delta x^2} \leq c$ for a fixed number $c > 0$. This equation is well-posed in the sense that there exists a maximum principle that determine the solution(see [OS88]). Marquina's fix also resolve the staircase quite well. We can also consider this equation as a level-set equation with normal speed $\nabla \cdot \frac{\nabla u}{|\nabla u|} + \lambda(u - u_0)$. But this model is not the exact gradient equation correspond to the original minimization problem.

As an important improvement of TV functional, Strong and Chan [Str97] introduced the weighted TV functional

$$TV_\alpha = \int_\Omega \alpha(x) |\nabla u(x)| dx$$

for spatially adaptive (selective) image restoration. The function $\alpha$ is chosen so that $\alpha$ is larger away from possible edges and smaller near a likely edge. Hence we allow for greater smoothing away from edges and less smoothing at the edges. Certain choices of $\alpha(x)$ were given in [CS96] and [Str97]. The numerical results are very promising. Chen and Wunderli [CW02] prove that there exists a unique solution for the weighted TV functional.

While keeping the sharping edges in images, the TV model suffers the staircase effect, namely, transform the smooth regions to piecewise constant regions. The reason is that the TV functional does not distinguish between jumps and smooth transitions. Several methods have been proposed to reduce the staircase effect. In [BC97], Blomgren, Chan and Mulet propose a "$TV - H^1$ interpolation" approach

that avoids the use of second order derivatives to reduce the staircase effect. Another approach by Chan, Marquina and Mulet [CMM00] is adding a nonlinear fourth order diffusive term to the Euler-Lagrange equations of the variational TV model. This technique substantially reduces the staircase effect, while preserving sharp jump discontinuities.

## 4.3 Adaptive TV Model

In this section, we will discuss the adaptive TV model and its properties. The adaptive TV model is:

$$\begin{aligned} \min_{u} \quad & \frac{1}{p}\int_{\Omega} |\nabla u|^p dxdy, 1 < p < 2, \\ \text{subject to} \quad & \frac{1}{2}\int (u-u_0)^2 dxdy = \sigma^2. \end{aligned} \tag{4.8}$$

The gradient decent flow of this equation is

$$u_t = \nabla \cdot (|\nabla u|^{p-2}\nabla u) - \lambda(u-u_0). \tag{4.9}$$

We have the following theorem:

**Theorem 2.** *For any $p > 1$, (4.9) has unique solution in $W^{1,p}(R^2)$.*

The proof can be found in [Kaw92]. To understand the limit behavior as $p \to 1$. We consider a special solution. Assume $\Omega = B_R(0)^N$ is a ball, then the solutions $u_p$ of the Dirichlet problem is

$$u_p(r) = (p-1)\frac{N}{p}\left[\left(\frac{R}{N}\right)^{p/p-1} - \left(\frac{r}{N}\right)^{p/p-1}\right],$$

and

$$u_p(r)/u_p(0) = 1 - (r/R)^{p/p-1}.$$

The shape of $u_p$ as $p \to 1$ is in Fig. (4.1).

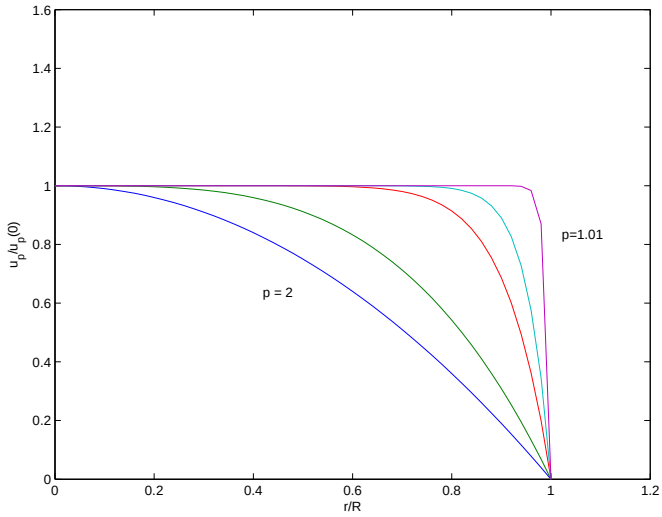



Figure 4.1: The shape of $u_p$ as $p \to 1$.

## 4.4 Numerical Implementation and Stability Properties

For simplicity, we consider a semi-discrete approximation to (4.9). We partition

$$R^2 = \cup \Omega_{ij},$$

where

$$\Omega_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{i-1/2}, y_{i+1/2}],$$

with

$$x_i = \frac{1}{2}(x_{i+1/2} + x_{i-1/2}), y_j = \frac{1}{2}(y_{j+1/2} + y_{j-1/2}).$$

As a measure of refinement, we call

$$\Delta = \max \left\{ \max_i (x_{i+1/2} - x_{i-1/2}), \max_j (y_{j+1/2} - y_{j-1/2}) \right\}.$$

Define the step function for each $t > 0$ as

$$u_\Delta(t) \equiv u_{ij}(t), x \in \Omega_{ij}.$$

For any step function, we define the difference operators based in equi mesh

$$D^x_{\pm}u_{ij} = \frac{\pm(u_{i\pm1,j} - u_{ij})}{\Delta x}, D^y_{\pm}u_{ij} = \frac{\pm(u_{i,j\pm1} - u_{ij})}{\Delta y}.$$

The approximation of $Z := |\nabla u|^{p-2}\nabla u$ can be expressed in terms of the difference operators for $u$:

$$Z_{1ij} = \frac{D^x_+ u_{ij}}{((D^x_+ u_{ij})^2 + (D^y_+ u_{ij})^2)^{\frac{2-p}{2}}}, \tag{4.10}$$

$$Z_{2ij} = \frac{D^y_+ u_{ij}}{((D^x_+ u_{ij})^2 + (D^y_+ u_{ij})^2)^{\frac{2-p}{2}}}. \tag{4.11}$$

$$\tag{4.12}$$

The semi-difference approximation of (4.9) are of the form

$$\frac{d}{dt}u_{ij}(t) = F_{ij}(t), i, j = 0, \pm1, \cdots, \tag{4.13}$$

where

$$F_{ij}(t) := D^x_- Z_{1ij} + D^y_- Z_{2ij}. \tag{4.14}$$

### 4.4.1 Numerical Implementation

We already have the spacial discretization. For time discretization, we can use either first order forward difference or high order Runge-Kutta discretization. Ino In our numerical experiments, we simply use the following scheme:

$$\begin{aligned} u^{n+1}_{i,j} = u^n_{i,j} + \frac{\Delta t}{h}\Bigg[\Delta^x_-\left(\frac{\Delta^x_+ u^n_{ij}}{((\Delta^x_+ u^n_{ij})^2 + (\Delta^y_+ u^n_{ij})^2)^{(2-p)/2}}\right) \\ + \Delta^y_-\left(\frac{\Delta^y_+ u^n_{ij}}{((\Delta^x_+ u^n_{ij})^2 + (\Delta^y_+ u^n_{ij})^2)^{(2-p)/2}}\right)\Bigg] \\ + \Delta t\, \lambda^n(u^n_{ij} - u_0(ih, jh)), \quad \text{for } i, j = 1, \cdots, N, \end{aligned} \tag{4.15}$$

where $h$ is the spacial step, $\Delta t$ be the tempal step.

### 4.4.2 Numerical Stability Properties

In this section, we will give various estimates for the scheme starting with given initial data of bounded variation.

**Lemma 3.** *Consider the semi-discrete scheme (4.9) subject to the initial data $u^0 \in l^2$, then*

$$\|u(t)\|_{L^p} \leq \|u_0\|_{L^p}. \tag{4.16}$$

*Proof.* By assumption, we have

$$\frac{d}{dt}u_{ij}(t) = D_-^x Z_{1ij} + D_-^y Z_{2ij}.$$

Then

$$\begin{aligned}
\frac{d}{dt}\sum |u_{ij}|^p &= p\sum |u_{ij}|^{p-1}\frac{du_{ij}}{dt} \\
&= p\sum |u_{ij}|^{p-1}\cdot (D_-^x Z_{1ij} + D_-^y Z_{2ij}) \\
&= -p\sum (Z_{1ij}\cdot D_+^x |u_{ij}|^{p-1} + Z_{2ij}\cdot D_+^y |u_{ij}|^{p-1}) \\
&= -p(p-1)\sum Z_{1ij}|u_{ij}|^{p-2}D_+^x u_{ij} + Z_{2ij}|u_{ij}|^{p-2}D_+^y u_{ij} \\
&= -p(p-1)\sum |u_{ij}|^{p-2}|\nabla u|^p \\
&\leq 0
\end{aligned}$$

Thus the $L^p$ norm is non-increasing. □

If we let $p \to \infty$, then we have:

**Lemma 4.** *For the semi-discrete scheme (4.9) subject to the initial data $u^0 \in l^2$, there exists a constant $C$ such that*

$$sup_{i,j}|u_{ij}(t)| \leq C, \forall t > 0.$$

**Lemma 5.** *Consider the semi-discrete scheme (4.9) subject to the initial data* $u^0 \in l^2$*, then*

$$\|\nabla u(t)\|_{L^p} \leq \|\nabla u_0\|_{L^p}. \tag{4.17}$$

*Proof.* Since

$$\begin{aligned}
\frac{d}{dt}\sum |\nabla u|^p &= \sum p|\nabla u_{ij}|^{p-2}[D_+^x u_{ij} \cdot \frac{d}{dt}u_{ij}(t) + D_+^y u_{ij} \cdot \frac{d}{dt}u_{ij}(t)] \\
&= p\sum(|\nabla u_{ij}|^{p-2}D_+^x u_{ij}) \cdot D_+^x F_{ij}) + (|\nabla u_{ij}|^{p-2}D_+^y u_{ij}) \cdot D_+^y F_{ij} \\
&= -p\sum[D_-^x(|\nabla u_{ij}|^{p-2}D_+^x u_i j) + D_-^y(|\nabla u_{ij}|^{p-2}D_+^x u_i j)F_{ij}(t) \\
&= -p\sum_{ij}(F_{ij}(t))^2 \\
&\leq 0.
\end{aligned}$$

Thus the $L^p$ norm of gradient of $u$ is non-increasing. □

## 4.5 Numerical Results

In this section, we will give the numerical results using our model on various synthetic and real images.

In Fig. 4.2 we show the result from TV denoising and Adaptive TV denosing on th 1-D signal. Fig. 1(c) clearly shows the unsatisfactory staircase effect on the smooth region, where in Fig. 1(d), we can see that the staircase effect has been reduced a lot and the sharp edge still be kept.

Fig.4.3(a) shows an 2-D image with smooth region. The image is generated by: $u(x, y) = 50+(y-1)/127*220$. Fig. 4.3(b) is the noise image. We can observe that in Fig.4.3(c) that there is a lot of block structure due to the staircase effect. Fig4.3 (d) show the Adaptive TV denoising with p = 1.5, the original image is perfectly recovered.

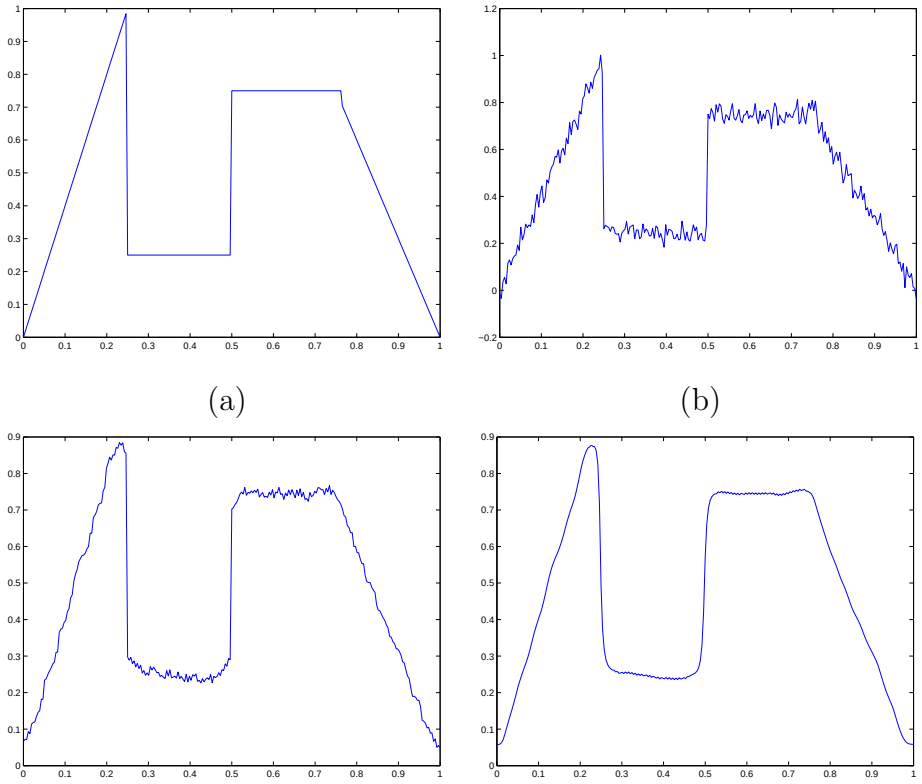
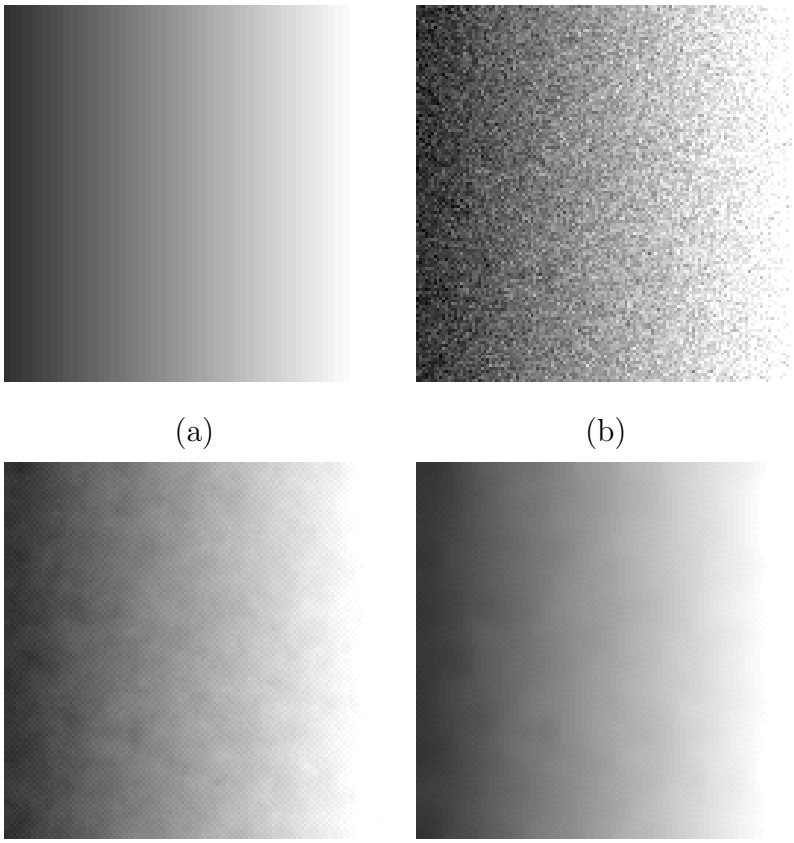


Figure 4.2: Comparison of TV and Adaptive TV on 1-D signal. (a)The original signal, (b) The noise signal, (c) signal after TV denoising, (d)Adaptive TV denoising with p = 1.4.



(a) (b)

(c) (d)

Figure 4.3: Comparison of TV and Adaptive TV on 2-D image. (a)Original image, (b) Noise image, (c)TV denoising, (d)Adaptive TV denoising with p=1.5.

In Fig.4.4 we show the results of Adaptive TV denoising with different p. The original image is white background with three vetical bars in the forground. Fig.4.4 (a) is the noisy image. Fig.4.4(b) is the result of TV denoising. Fig.4.4(c) and (d) show the result of Apaptive TV with $p = 1.25$ and $p = 1.6$ respectively. As we expected, when p is close to 2, the edges become smeared because of that when $p = 2$, the model is just least square model which notorious for destroy the edeges in the image. we can also see the the staircase effect is alleviated in Fig.4.4 (c) as compared to Fig.4.4(b).

Fig.4.5 and Fig.4.6 show another comparison between the TV model and Adaptive TV model. We can see that in TV model the recovered image is not satisfactory because of the staircase effect. We also see that when $p$ is close to 1, the edge is as well preserved as those of the TV model and the staircase effect is reduced.

## 4.6 Concluding Remarks

In this chapter, we propose an extension of Total Variation denoising model. This new model can effectively reduce the staircase effect in TV model whereas it can still keep the sharp edge as those of TV model. We prove some stability properties of the corresponding Euler-Lagrange equation. Various numerical examples validate our model.

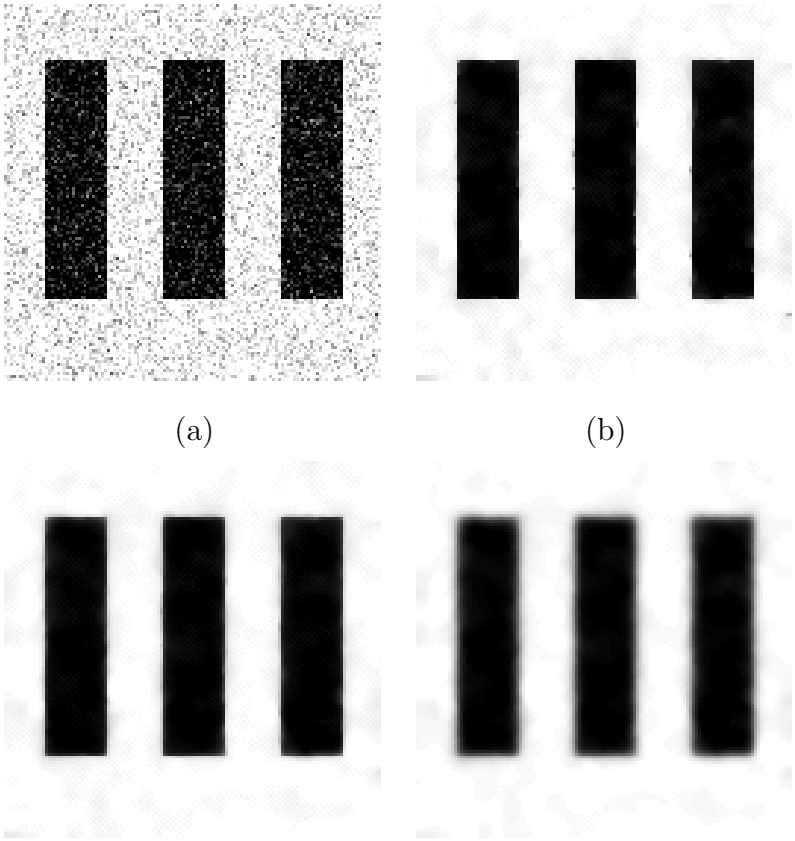
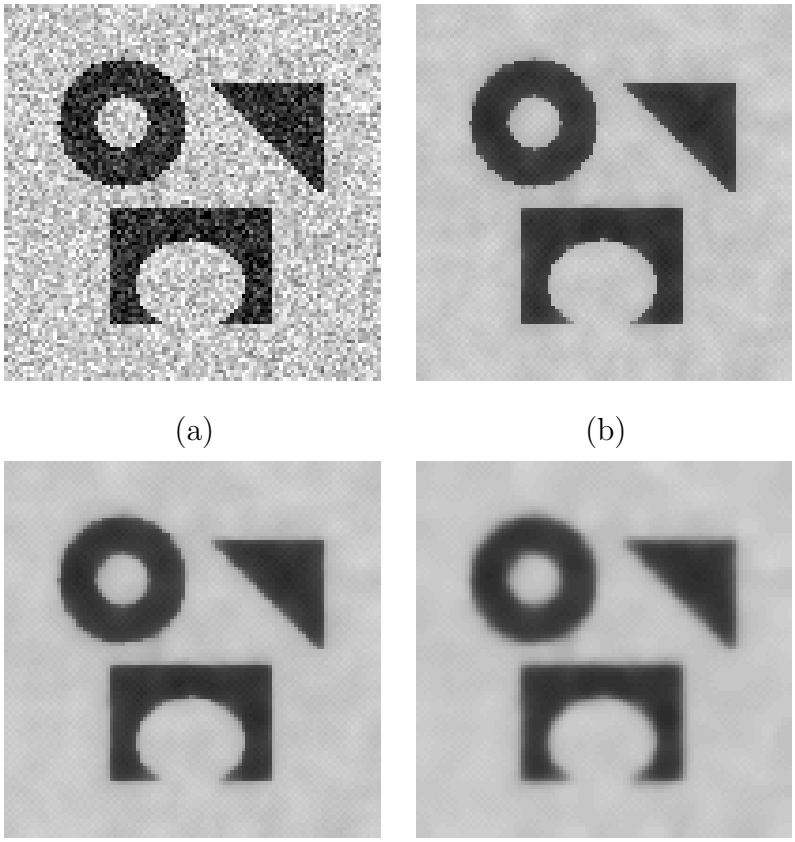


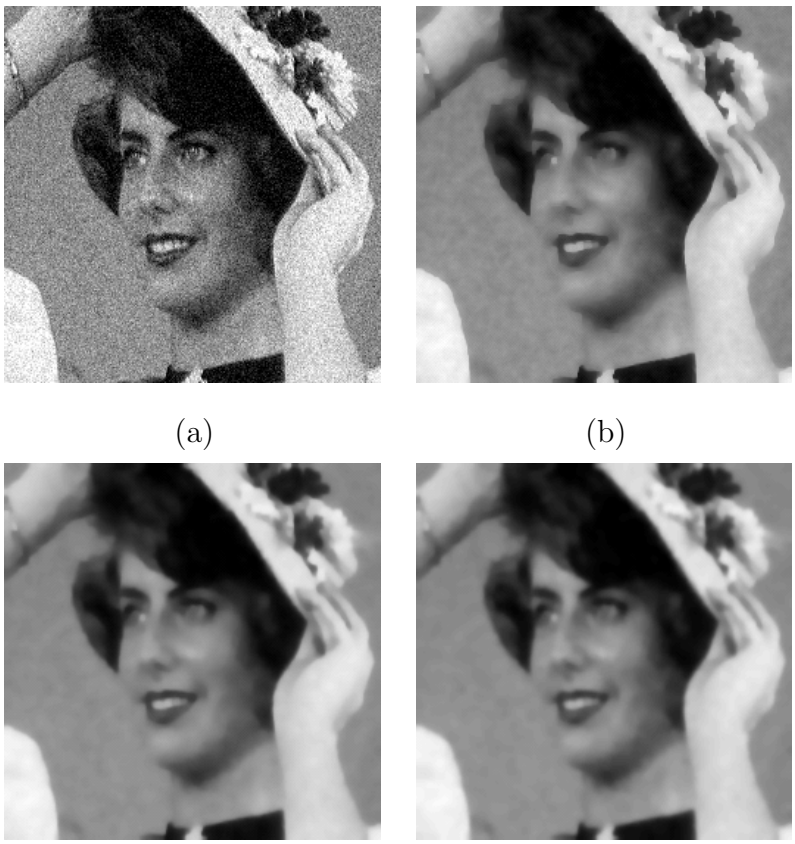
Figure 4.4: Comparison of TV and Adaptive TV on 2-D image. (a)Noise image, (b) TV denoising, (C)Adaptive TV denosing with p = 1.25, (d) Adaptive TV denoising with p = 1.6.



(a) (b)

(c) (d)

Figure 4.5: Comparison of TV and Adaptive TV on 2-D image. (a)Noise image, (b) TV denoising, (C)Adaptive TV denosing with p = 1.25, (d) Adaptive TV denoising with p = 1.5.



(a) (b)

(c) (d)

Figure 4.6: Comparison of TV and Adaptive TV on 2-D image. (a)Noise image, (b) TV denoising, (C)Adaptive TV denosing with p = 1.4, (d) Adaptive TV denoising with p = 1.6.

# References


[AGL93] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel. "Axioms and fundamental equations of image prodessing." *Arch. Ration. Mech. and Anal.*, **123**:199–257, 1993.

[Ash01] M. Ashikhmin. "Synthesizing Natural Textures." In *The proceedings of 2001 ACM Symposium on Interactive 3D Graphics*, pp. 217–226, March 2001.

[AV94] R. Acar and C. R. Vogel. "Analysis of total variation penalty methods for ill-posed problems." *Inverse Problems*, **10**:1217–1229, 1994.

[BBC01] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. "Filling-in by joint interpolation of vector fields and gray levels." *IEEE Trans. on Image Processing*, **10**(8):1200–1211, August 2001.

[BC97] P. V. Blomgren and T. F. Chan. "Modular Solvers for Constrained Image Restoration Problems." *Technical report, UCLA Dept. of Math.*, **CAM 97-52**, 1997.

[BCO01] Marcelo Bertalmío, Li-Tien Cheng, Stanley Osher, and Guillermo Sapiro. "Variational problems and partial differential equations on implicit surfaces." *J. Comput. Phys.*, **174**(2):759–780, 2001.

[Br99] Pierre Brémaud. *Markov Chains: Gibbs fields, Monte-Carlo simulation, and queues.* Springer, 1999.

[BSB99] M. Bertalmio, G. Sapiro, C. Ballester, and V. Caselles. "Image Inpainting." *University of Minnesota IMA TR*, December 1999.

[BSI87] J. Beck, A. Sulter, and R. Irvy. "Spatial frequency channels and perceptual grouping in texture segmentation." *CVGIP: Image Understanding*, **37**:299–325, 1987.

[BVS02] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. "Simultaneous Structure and Texture Image Inpainting." *UCLA CAM Report 02-47*, July 2002.

[CKS97] V. Caselles, R. Kimmel, and G. Sapiro. "Geodesic active contours." *International Journal of Computer Visison*, **22**(1):61–79, 1997.

[CKS02] Tony F. Chan, Sang-Ha Kang, and Jianhong Shen. "Euler's elastica and curvature-based image inpainting." *SIAM J. Appl. Math*, **63**(2):564–592, 2002.

[CMM00] Tony Chan, Antonio Marquina, and Pep Mulet. "High-order total variation-based image restoration." *SIAM J. Sci. Comput.*, **22**(2):503–516 (electronic), 2000.

[Coh91] D. Cohen. "On active contour models and ballons." *CVGIP: Image Understanding*, **53**:211–218, 1991.

[CS96] T. Chan and D. Strong. "Relation of regularization parameter and scale in total variation based image denosing." *Technical report, UCLA Dept. of Math.*, **CAM 96-7**, 1996.

[CS01] T. F. Chan and J. Shen. "Non-texture inpainting by curvature-driven diffusions (CDD)." *J. Visual Comm. Image Rep.*, **12**(4):436–449, 2001.

[CS02] T. F. Chan and J. Shen. "Mathematical models of local non-texture inpaintings." *SIAM J. Appl. Math*, **62**(3):1019–1043, 2002.

[CV00a] T.F. Chan and L.A. Vese. "Image segmentation using level sets and the piecewise constant Mumford-Shah model." *Technical report, UCLA Dept. of Math.*, **CAM 00-14**, 2000.

[CV00b] T.F. Chan and L.A. Vese. "A level set algorithm for minimizing the Mumford-Shah functional in image processing." *Technical report, UCLA Dept. of Math.*, **CAM 00-13**, 2000.

[CV01] T.F. Chan and L.A. Vese. "Active contours without edges." *IEEE Trans. on Image Processing*, **10**(2):266–277, February 2001.

[CW02] Y. Chen and W. Wunderli. "Adaptive total variation for image restoration in BV space." *J. Math. Anal. Appl*, **272**:117–137, 2002.

[DeB97] J.S. DeBonet. "Multiresolution sampling procedure for analysis and synthesis of texture images." In *Proc. of SIGGRAPH97*, pp. 361–368, 1997.

[EL99] A. Efros and T. Leung. "Texture Synthesis by Non-parametric Sampling." In *International Conference on Computer Vision 2*, pp. 1033–1038, September 1999.

[Fed01] R. Fedkiw. "Private communication during the seminar given by the second author at Stanford University." April 2001.

[GF02] F. Gibou and R. Fedkiw. "A fast level set based algorithm for segmentation." *Submitted to CVPR, private communication*, November 2002.

[HB95] D. Heeger and J. Bergen. "Pyramid-based texture analysis/synthesis." *Computer Graphics (SIGGRAPH 1995)*, pp. 229–238, July 1995.

[Kaw92] B. Kawohl. "Remarks on the operator $div(\nabla u/|\nabla u|$." *Contemporary Mathematics*, **127**:69–83, 1992.

[KLM94] G. Koepfler, C. Lopez, and J.M. Morel. "A multiscale algorithm for image segmentation by variational method." *SIAM J. Numer. Anal.*, **31**(1):282–299, February 1994.

[KWT87] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes: Active Contour Modles." *International Journal of Computer Visison*, **1**(4):321–331, 1987.

[MM98] M. Masnou and J. Morel. "Level-lines based disocclusion." In *IEEE Int. Conf. on Image Processing*, pp. 259–263, October 1998.

[MO00] A. Marquina and S. Osher. "Explicit algorithms for a new time dependent model based on level set motion for a nonlinear debluring and noise removal." *SIAM J. Sci. Comput.*, **22**(2):387–405, 2000.

[MS89] D. Mumford and J. Shah. "Optimal approximation by piecewise smooth functions and associated variational problems." *Comm. Pure Appl. Math.*, **42**:577–685, 1989.

[MS95] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation*, volume 14 of *Progress in Nonlinear Differential Equations and Their Applications*. Birkhäuser, Boston, 1995.

[NMS93] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, Segmentation, and Depth*. Springer-Verlag, 1993.

[OS88] S. Osher and J.A. Sethian. "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation." *J. Comput. Phys.*, **79**:12–49, 1988.

[Par00] Nikos Paragios. *Geodesic Active Regions and Level Set Methods: Contributions and Applications in Artifical Vison*. PhD thesis, 2000.

[PD98] N. Paragios and R. Deriche. "Geodesic Active Regions for Supervised Texture Segmentation." *INRIA*, **RR-3440**, June 1998.

[PD99] N. Paragios and R. Deriche. "Coupled Geodesic Active Regions for Image Segmentation." Technical Report RR-3783, INRIA, October 1999.

[PD00] N. Paragios and R. Deriche. "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects." *PAMI*, **22**(4):415, April 2000.

[Pm90] P. Perona and J. malik. "Scale-space and edge detection using anisotropic diffusion." *IEEE Trans. Pattern Anal. and Mach. Intell.*, **12**:629–639, 1990.

[PM97] T. F. Chan P. V. Blomgren and P. Mulet. "Extensions to Total Variation Denosing." *Technical report, UCLA Dept. of Math.*, **CAM 97-42**, 1997.

[PP93] K. Popat and R. Picard. "Novel cluster-based probability model for texture synthesis, classification and compression." In *Visual Communications and Image Processing*, pp. 756–768, 1993.

[ROF92] L. Rudin, S. Osher, and E. Fatemi. "Nonlinear Total Variation based Noise Removal Algorithm." *Physica D*, **60**:259–268, 1992.

[SP98] E.P. Simomcelli and J. Portilla. "Texture characterization via joint statistics of wavelets coefficient magnitudes." In *Proc. 5th Int. Conf. on Image Processing*, volume 1, pp. 62–66, October 1998.

[Str97] D. Strong. *Adaptive total variation minimization image restoration.* PhD thesis, 1997.

[TM99] G. Golub T. Chan and P. Mulet. "A nonlinear primal-dual method for total variation-based image restoration." *SIAM J. Sci. Comput.*, **20**:1964–1977, 1999.

[TYW01] A. Tsai, A. Yessi, and A.S. Willsky. "Curve Evolution Implementation of the Mumford-Shah Functional for Image Segmentation, Denosing, Interpolation and magnification." *IEEE Trans. on Image Processing*, **10**(8):1169–1186, August 2001.

[TZ02] Z.W. Tu and S.C. Zhu. "Image Segmentation by Data Driven Markov Chain Monte Carlo." *IEEE Trans on Pattern Analysis and Machine Intelligence*, **24**(5):657–673, May 2002.

[Ves02] L.A. Vese. "Multiphase Object Detection and Image Segmentation." *Technical report, UCLA Dept. of Math.*, **CAM 02-36**, 2002.

[VO96] C.R. Vogel and M.E. Oman. "Iterative methods for total variation denosing." *SIAM J. Sci. Comput.*, **17**:227–238, 1996.

[WL00] Li-Yi Wei and Marc Levoy. "Fast Texture Synthesis Using Tree-Structured Vector Quantization." In *Proceedings of SIGGRAPH 2000*, pp. 479–488, July 2000.

[ZWM98] S.C. Zhu, Y.N. Wu, and D. Mumford. "Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for texture Modeling." *Internatinal Journal of Computer Visison*, **27**(2):107–126, March 1998.

[ZY96] S.C. Zhu and A. Yuille. "Region Competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation." *IEEE Trans on Pattern Analysis and Machine Intelligence*, **18**(9):884–900, September 1996.